

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2002 年 5 月 30 日 (30.05.2002)

PCT

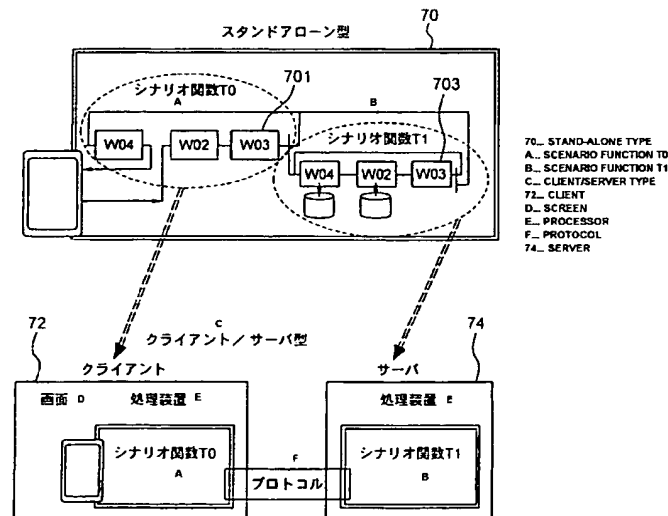
(10) 国際公開番号
WO 02/42904 A1

- (51) 国際特許分類⁷: G06F 9/06, 9/44 エスデー研究所 (INFORMATION SYSTEM DEVELOPMENT INSTITUTE) [JP/JP]; 〒108-0074 東京都港区高輪三丁目11番3号 Tokyo (JP).
- (21) 国際出願番号: PCT/JP01/10254
- (22) 国際出願日: 2001 年 11 月 22 日 (22.11.2001) (72) 発明者; および (75) 発明者/出願人 (米国についてのみ): 根来文生 (NEGORO, Fumio) [JP/JP]; 〒248-0001 神奈川県鎌倉市十二所 967-64 Kanagawa (JP).
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語 (74) 代理人: 友野英三 (TOMONO, Eizo); 〒184-0011 東京都小金井市東町4丁目42番地11号-B103 サクラ国際特許事務所 友野オフィス Tokyo (JP).
- (30) 優先権データ:
特願 2000-358405
2000 年 11 月 24 日 (24.11.2000) JP (81) 指定国 (国内): AU, CA, CN, IN, JP, KR, NZ, RU, SG, US.
- (71) 出願人 (米国を除く全ての指定国について): ソフトウェア生産技術研究所株式会社 (THE INSTITUTE OF COMPUTER BASED SOFTWARE METHODOLOGY AND TECHNOLOGY) [JP/JP]; 〒108-0074 東京都港区高輪三丁目11番3号 Tokyo (JP). 株式会社 アイ (84) 指定国 (広域): ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR).
- 添付公開書類:
— 国際調査報告書

[続葉有]

(54) Title: SOFTWARE SYNCHRONOUS SPECIFYING METHOD, VALIDATING METHOD, VERIFYING METHOD, AND TESTING METHOD

(54) 発明の名称: ソフトウェアの同期式スペシファイング方法、バリデーティング方法、ペリファイング方法、及びテスト方法



(57) Abstract: According to the invention, only by directly implementing the law itself of a scenario function in which a mechanism for determining the significance of a word by an infinite repetition structure subsists, a result equivalent to the one obtained by implementing infinite repetition of an activity such as specifying, validating, verifying, or testing necessary for producing a perfect product (software desired by a development requester) by a conventional method is obtained, thereby avoiding such infinite repetition and providing software for completely meeting a demand for development.

[続葉有]



WO 02/42904 A1



2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約:

本発明では、無限回の繰り返し構造によって単語の有意性を決定する仕組みを内在しているシナリオ関数の法則そのものをプログラミング言語で直接的にインプリメントすることだけで、従来法で完璧な製品（開発要望者所望のソフトウェア）にするために必要とされるスペシファイイング、バリデーティング、ペリファイイング及びテストイングというアクティビティの無限回の繰り返しを実施したと同等の成果が達成され、当該無限回の繰り返しを回避しつつ開発要望を完璧に実現するソフトウェアの提供が実現される。

1

明 細 書

ソフトウェアの同期式スペシファアイイング方法、バリデーティング方法、ベリファイイング方法、及びテストイング方法

5

技術分野

本発明は、例えば業務用・個人用を問わずあらゆるソフトウェアに係るソフトウェアの同期式スペシファアイイング (specifiing) 方法、バリデーティング (validating) 方法、ベリファイイング (verifiing) 方法、
10 及びテストイング (testing) 方法に関する。

背景技術

従来より、処理装置のさまざまな構成、例えば、スタンドアローン型、クライアント／サーバ型、クライアント／フロントサーバ／アプリケーションサーバ型に対して適用業務や適用機能をソフトウェアとしてインプリメント（実現）するに際しては、適用業務や適用機能が潜在的に発生させ得るさまざまな要求事象と、処理装置のさまざまな構成から誘導され得るさまざまな処理事象との間で生ずるさまざまなコンフリクト（衝突）事象に対し、矛盾なく対処する仕組みを確立し、単語を要素と
15 する文章で表さなければならない。

しかしながら、このような要求事象や処理事象とそれらとの間に生ずるコンフリクト事象とそれに対する対処方法のすべてを人間技で抽出するということは原理的に不可能である。

なんとなれば、60兆の細胞という有限な要素を基にして要求事象や
25 処理事象やコンフリクト事象という現象に対して認識を成立させ、その

2

現象を説明するための文章を作成するに必要な文章の要素である単語そのものを心の中で決定した後、具体的な言葉の発言という言動を表す人間という存在が、その要素の数（60兆の細胞）と比較して無限に近い数の要素（この要素は、例えば宇宙全体を構成する素粒子とその素粒子相互から導出される現象）から成る要求事象や処理事象やコンフリクト事象の総てを捉えることは、“小は大よりも小さいのだから、小さいものがそれよりも大きいものは捉えることはできない”という数学的原理からも不可能であるからである。

5 5
10 すべての事象を人間技で捉えることはできないということから、ソフトウェア開発の当事者には、

- ・スペシファイイング（仕様定義）、
- ・バリデーティング（その仕様の正統性の確認）、
- ・ベリファイイング（その仕様記述の間違い排除）、

及び

15 ・テストイング（仕様どおりに作られているかのテスト）

というアクティビティ（作業）の実行によって、捉えられる範囲の要求事象や処理事象やコンフリクト事象を捉え、その範囲内で相互に矛盾のない対処方法を模索し、かかる模索結果を目的ソフトウェアとしてインプリメントしなければならないことになる。

20 このようなアクティビティの本質はとりも直さず、要求事象や処理事象やコンフリクト事象や対処方法の説明のため、文法によって単語を部分集合させた文章を作り、それにより説明するということに他ならない。

この単語とは自然言語を構成する要素である。その単語を部分集合化させた文章で漏れや齟齬のないように説明するためには、その構成要素である単語自身の選択がその事象を厳密に説明するにふさわしい単語であって、且つ、漏れのないようになっていなければならない。その為に

25

3

は、その単語を使おうとした意図が明確でなければならないし、その単語の意味をはっきりと理解していなければならない。

しかし、単語を使おうとした意図は人のこころの中に埋没しており、他人はおろか当人にも直接的には決してうかがい知れる事柄ではありえない。直接的にうかがい知れないことから、その意図をはっきりと捉える為には無限回の自問自答や質疑応答が必要となるのである。

また、単語の意味をはっきりと理解するために辞書をひいて理解しようとする、辞書には数多くの説明文章があり、かつ、その説明文章も単語の集合であることから、当該単語の正確な意味を捉えることは至難の技となる。すなわち、単語の意味を理解しようとして辞書を引くと、そのひとつの単語を説明するために必要な単語は更に50倍から100倍も必要ということになる。その50倍から100倍の単語の意味をそれぞれ理解しようとして辞書を引くと更に結果として50倍にも100倍にも単語が膨らむ。これを繰り返していくと辞書を無限回調査して無限個の単語についてその意味を調べなければ、最初の単語の正確な意味は捉えられないことになる。しかし、無限というのは限界が定まらないということであるから、結局、単語の意味は無限であり不定ということになる。(このようなことは集合論において、ラッセルのパラドックスとかゲーデルの不確定性定理といわれてもいる)。

結論として、「単語の意味を理解しようとして開発要望（単語の集合）の意図を満たそうとするアプローチ」は、集合論におけるラッセルのパラドックスを回避できない限り、原理的に不可能ということになる。

つまり、完璧な製品をインプリメントするために前述のアクティビティが必要とされているのところ、そのアクティビティということを持立させる要素が単語の意図と意味なのであり、その単語の意図と意味を完璧にするためには無限回に近い回数分の自問自答や質疑応答や調査が必

4

要であるということになるが、無限回に近いこのようなアクティビティの繰り返しは現実的に実行不可能である。

そこで、現実的には、限定的な回数のアクティビティに止め、その限定的な回数による成果として纏められた限定的な範囲の仕様に対し、開発要望者と技術者とが暫定的に合意するという手続きをソフトウェアにかか

5 かる従来技術では採用せざるを得なかった。

この結果、従来においては、完璧な製品は無限回のアクティビティの繰り返しを行わなければ実現できないことから、限定的な回数のアクティビティでお茶を濁さざるを得なかったこれまでのソフトウェア開発方法との間には「お客様満足度（CS）」の評価尺度とされているQ

10 (quality), C(cost), D(delivery)そのものに原理的な限界と矛盾が内在していたことになる。

このように、従来のソフトウェア開発には、開発要望のソフトウェア実現において、そもそも開発要望の捉え方という出発点からして原理的な限界と矛盾が内在するという問題があった。

15

発明の開示

本発明は上記の従来技術の問題を解決するためになされたもので、従来の原理的な限界と矛盾を解消することを目的とする。

20 本発明の更なる目的は、完璧な製品（開発要望者所望のソフトウェア）にするために必要とされるスペシファイイング、バリデーティング、ベリファイイング及びテストイングというアクティビティの無限回の繰り返しを実施したと同等の成果を実際のかかるアクティビティなしに達成することにある。

25 かかる課題を解決するため本発明は、生産するシステムに係る画面に表示するための送信情報を決定する第1のステップと、前記送信情報を

5

編集するW 0 4パレットの識別子を記録しているデータ・フィールドのデータコードを検査する第2のステップと、前記第2のステップの結果、画面に表示するための送信情報がメニューである場合には当該システム内の総てのパレットを起動対象としてセットする第3のステップと、前記第2のステップの結果、当該画面に表示するための送信情報がメニューでない場合には必要（新規／継続の指定）に応じて使用済みデータ・フィールドを初期化する第4のステップと、前記送信情報を編集するW 0 4パレットを起動する第5のステップと、前記第5のステップにおいてW 0 4パレットにより編集されたデータコードを当該画面に送信する第6のステップと、前記画面上で操作されたデータコードの入力及びその処理指令とを受信する第7のステップと、前記受信されたデータコード及び処理指令を受け入れるW 0 2パレットを起動する第8のステップと、必要に応じてW 0 3パレット或いはT 1にかかるパレット連鎖関数を起動する第9のステップと、前記第1のステップ乃至第9のステップをシステム閉塞に至るまで無限に繰り返す第10のステップとを具備することを特徴とする。

「パレット連鎖関数」とは、本発明に係る理論的帰結であるシナリオ関数の構成要素の一つであり、同構成要素のW 0 2パレット関数、W 0 3パレット関数、及び、W 0 4パレット関数を起動する働きを持ち、かつ、その論理構造（プログラムとして実現された場合には、プログラミング構造）が、生産対象のソフトウェアの機能・種類に拘らずに所与であるものをいう。

上記の構成を備える本発明では、無限回の繰り返し構造によって単語の有意性を決定する仕組みを内在しているシナリオ関数の法則そのものをプログラミング言語で直接的にインプリメントすることだけで、従来法で完璧な製品（開発要望者所望のソフトウェア）にするために必要と

6

されるスペシファイニング、バリデーティング、ベリファイイング及びテストイングというアクティビティの無限回の繰り返しを実施したと同等の成果が達成される。

5 図面の簡単な説明

第1図は、本発明に係るシナリオ関数のコンポーネントであるパレット連鎖関数の論理構造を説明するためのフローチャートである。

第2図は、本発明に係るシナリオ関数のコンポーネントであるパレット関数の論理構造を説明するためのフローチャートである。

- 10 第3図は、本発明に係るシナリオ関数のコンポーネントである基底論理（論理要素）の論理構造を説明するためのフローチャートである。

第4図は、本発明の一実施形態に係る単語が集合化された文章という開発要望と処理装置との関係を説明する概念図である。

- 15 第5図は、本発明の一実施形態に係るシナリオ関数の三つめのコンポーネントである単語ごとの基底論理が生命作用を司るニューロンとみなした場合の生命作用と生命現象との関係を説明するブロック図である。

第6図は、本発明の一実施形態に係るインプリメントされたプログラムの構成と作動の順序を説明するブロック図である。

- 20 第7図は、本発明の一実施形態に係る適用業務や適用機能を二つの処理装置がクライアントとサーバ型とに分かれて執行するソフトウェアをシナリオ関数の立場で説明する概念的ブロック図である。

第8図は、本発明に係る双方の生命体同士の間でひとつの目的を達成するに必要となるプロトコルを説明するためのシーケンシャル・チャートである。

- 25 第9図は、本発明の一実施形態に係るプロトコルに係る単語を、シナリオ関数 T 1 を具現化するプログラムである各パレット W 0 4、W 0 2、

W 0 3 中の定型的な（変数）識別子挿入位置に貼り付ける様子を説明する概念的ブロック図である。

第 1 0 図は、本発明の一実施形態に係る適用業務や適用機能を三つの処理装置がクライアントとフロントサーバとアプリケーション・サーバ
5 とに分かれて執行するソフトウェアをシナリオ関数の立場で説明するための概念的ブロック図である。

第 1 1 図は、本発明の一実施形態に係る考え方を概念的に表したコンセプチュアル・チャートである。

10 発明の概要

本発明の基本となる考え方は、L y e e と呼ばれる開発方法論であり、これについては、P C T / J P 9 6 / 0 3 1 8 3、P C T / J P 9 7 / 0 1 4 9 2、P C T / J P 9 9 / 0 1 3 9 4、P C T / J P 0 0 / 0 4 0 0 8、特願平 1 1 - 3 2 1 7 8 8 号、及び、特願平 1 2 - 1 6 0 8 8
15 7 号で開示されているものである。

L y e e と呼ぶ開発方法論において単語の有意性を決定する内面の法則として確立されているシナリオ関数は、次の式で表される。

$$\begin{aligned}
 T 0 &= \Phi 0(+\{\Phi p \{ \quad, L 2(R,i), P 2(R,G), P 2(R,j) \} \} R \\
 &\quad +\{\Phi p \{ Y 3(R,i), R 3(B,i), L 3(R,i), L 3(S,i), P 3(j), T 1 \} \} \\
 20 &\quad +\{\Phi p \{ Y 4(S,i), L 4(S,i), P 4(S,P), P 4(S,j) \} \} S) \\
 T 1 &= \Phi 1(+\{\Phi p \{ \quad, L 2(r,i), P 2(r,G), P 2(j) \} \} \\
 &\quad +\{\Phi p \{ Y 3(r,i), R 3(b,i), L 3(r,i), L 3(w,i), P 3(j) \} \} \\
 &\quad +\{\Phi p \{ Y 3(w,i), L 4(w,j), P 4(w,P), P 4(j) \} \})
 \end{aligned}$$

ここで、「T 0」は、単語が属する媒体を、識別子 R (Receive) にて指標される入力画面、及び、識別子 S (Send) にて指標される出力画面とするシ
25 ナリオ関数であり、「T 1」は、「T 0」にかかる単語の有意性決定を補

助する単語が属する媒体を、識別子 r (read)/にて指標される入力ファイル、及び、識別子 w(write)にて指標される出力ファイルあるいは帳票とするシナリオ関数である。

- 5 シナリオ関数の右辺は、その単語の「意味になる何か」であるところの有意性であるデータコードをその媒体上のその単語で指標されるデータ・フィールドに具象化するプログラムである。

- 10 なお、「意味」とは、その単語で識別されるデータ・フィールドに具象化されたデータコードを見た人の心（マインド）に湧出した何か（言表不可能なもの）であると考ええる。そして、その「何か」とは、人の心で知覚され、その知覚を契機として人（生命体）が生命の続く限りそれに対して反応（言動などの生命現象）させる原因に相当する事柄と考える。

- 15 するとかかる「反応」こそ、完璧なプログラム製品にするために必要であるとされているスペシファイイング、バリデーティング、ベリファイイング及びテストイングというアクティビティそのものとみなせることになる。

その反応の回数を n とすると、スペシファイイング、バリデーティング、ベリファイイング及びテストイングというアクティビティを経て実現されたプログラム製品が果たす機能 P_A とは、

$$P_A \equiv T 0^n, P_A \neq T 0$$

- 20 と表記できる。

- 25 即ち、L y e e の方法で実現されるプログラム $T 0$ は「 n 回分のアクティビティ（即ち、完璧なプログラム製品にするために必要であるとされているスペシファイイング、バリデーティング、ベリファイイング及びテストイング）を経たものではない」のであるが、「その同じプログラム（シナリオ関数の右辺）を n 回繰り返したものが要望を満たすための機能を果たすプログラム製品である」ということでもある。 n 回繰り返

すプログラムが常時同じ T_0 であるということは、プログラム T_0 は個々のアクティビティ毎に変わることないということでもある。それ故、結果としてプログラム T_0 は n 回分のアクティビティを繰り返した状態に至っているということがいえる。

- 5 そのシナリオ関数の右辺のうち、「Φ 0(+R+S)」並びに「Φ 1(+R+S)」をパレット連鎖関数と呼び、その構造法則をプログラミング言語で記述すればプログラムとなる。

「 $\Phi_p \{,,, \}$ 」はパレット関数と呼び、その構造法則をプログラミング言語で記述すればプログラムとなる。

- 10 「Yn()」, 「Rn()」, 「Ln()」, 「Ln()」, 「Pn()」は総称して基底論理と呼び、その構造法則をプログラミング言語で記述すればプログラムとなる。

その内、

「 $Y_n()$ 」は位相要素、

- 15 「R_n()」はR型位相要素、

「Ln()」は、

「L3(R,i)」、「L3(r,i)」を位相要素型論理要素；

「L2(R,i)」、「L3(S,i)」、「L4(S,i)」、「L2(r,i)」、「L3(w,i)」、「L4(w,i)」は論理要素、

- 20 「P_n()」は総称して作用要素、

その内、「P 2(G)」,「P 4(P)」を指令要素と、「P n(j)」を経路要素と構造要素と呼ぶ。

ところで、単語とはソフトウェアの開発要望として開発要望者が自然言語で発した意味を有することになる情報の最小の単位となるものに対応する。

一方、シナリオ関数とは開発要望者がソフトウェアの開発要望として

その単語をその単語として発言しようとした心の中で決めた意図(つまり、これが意味なのであるが)をその単語自身を変数として求める関数でもある。

即ち、 $y = f(x)$ で表す関数の場合、変数は x であるように、シナリオ関数の場合の変数は単語 i 、より厳密には単語 i に係る識別子である（以下では、便宜上、「シナリオ関数の場合の変数は単語 i である」といういい方もする）。

即ち、

$$T0 = \Phi 0(+\{\Phi p \{Ln(\text{単語 } i), \dots\}\})$$

10 である。

このとき、開発要望者がソフトウェアの開発要望を単語の部分集合の文章として発した「局面」を、情報処理システムの場合は「画面」など人が5感で直接的に認識できる媒体とみなしたのである。従ってかかる「局面」は、必ずしもいわゆる「画面」だけに限定するのではなく人が5感で認識できる媒体であればよい。

そのような媒体を識別する識別子 R 、 S 、 r 、 w を定義体識別子と呼び、また、それらの媒体上にペースト（貼り付け）された単語を識別する識別子 i を単語体識別子と呼び、作用要素の場合はその作用要素を識別する識別子 G 、 P 、 j を作用子と呼ぶ。

20 このシナリオ関数におけるパレット連鎖関数の「 $\Phi 0(+R +S)$ 」並びに「 $\Phi 1(+R +S)$ 」において、「 $($ 」は定義体からのデータのGET機能（受信機能やREAD機能）に対応する。

一方、‘)’は定義体へのデータのPUT機能（送信機能やWRITE機能）に対応し、‘+’はプログラムの起動機能に対応する。‘{ }D’は
25 { }内の要素を定義体識別子Dに属す分だけ集めるという意味であり、具体的にはDで指標されるプログラムを起動の対象に含める という意味

である。

なお、T1 の場合の添え字「2」、「4」で集められるグループ、及び T0 及び T1 の場合の添え字「3」で集められるグループについては D が添字になっていないから、それぞれのグループのプログラムは 1 本だけである。このようなグループをパレットと呼び、添え字「2」で集められるグループを W02 パレット、添え字「3」で集められるグループを W03 パレット及び添え字「4」で集められるグループを W04 パレットと呼ぶ。

またシナリオ関数におけるパレット関数の「 $\Phi p\{\dots\}$ 」において、「 $\{\}$ 」は $\{\}$ 内の要素即ち基底論理を集めるという意味である。その集める範囲は $\langle L$ (第 1 の変数、第 2 の変数) \rangle で表される各基底論理の第 1 の変数に対応する定義体に属する第 2 の変数に対応する単語ごとのプログラム分である。この集めるということを CALL 命令や PERFORM 命令で代替させてもよい。

第 1 図乃至第 3 図は、本発明に係るシナリオ関数のコンポーネントであるパレット連鎖関数、パレット関数及び基底論理（論理要素）の論理構造を説明するためのフローチャートである。

すなわち、同図に示すようにパレット連鎖関数においては、まず、システムに係る画面に表示するための送信情報を決定する（ステップ 101）。

次に、その送信情報を編集する W04 パレットの識別子を記録しているデータ・フィールドのデータコードを検査する（ステップ 102）。

かかる第 2 のステップの結果、かかる画面に表示するための送信情報がメニューである場合には当該システム内の総てのパレットを起動対象としてセットする（ステップ 103）。

ステップ 102 の結果、かかる画面に表示するための送信情報がメニ

ユーでない場合には必要(新規／継続の指定)に応じて使用済みデータ・フィールドを初期化する(ステップ104)。

次に、その送信情報を編集するW04パレットを起動する(ステップ105)。

- 5 続いてステップ105においてW04パレットにより編集されたデータコードを当該画面に送信する(ステップ106)。

この結果、かかる画面に対して利用者が操作し終わることを待ち(ステップ107)、かかる画面上で操作されたデータコードの入力及びその処理指令とを受信する(ステップ108)。

- 10 次に、かかる受信されたデータコード及び処理指令を受け入れるW02パレットを起動する(ステップ109)。

必要に応じてW03パレット或いはT1にかかるパレット連鎖関数を起動する(ステップ110)。

- そしてこれらのステップ101乃至ステップ110をシステム閉塞に至るまで無限に繰り返す(ステップ111)。
- 15

次に、パレット関数は、パレット連鎖関数のステップ106で送信される情報を表示する画面ごとに設けられるW04パレット、同じくパレット連鎖関数のステップ108で受信される情報が存在していた画面ごとに設けられるW02パレット及びシステムに1つだけ設けられるW0

- 20 3パレットに実装される。

第2図に示すようにパレット関数は、シナリオ関数の三つ目のコンポーネントであるところの基底論理を「その単語の有意性が成立する(再起動フラグがリセットされる)まで無限に起動を繰り返す」ステップを具備することが特徴的である。

- 25 最後に、基底論理(論理要素)は、画面に属する単語(この単語は開発を要望された適用業務や適用機能を説明する文章を構成している単語

そのものである) ごとに設けられ、当該画面に対応するW 0 4パレットとW 0 2パレット及びW 0 3パレットに実装される。この基底論理は、前述したパレット関数によって「その単語の有意性が成立する(再起動フラグがリセットされる)まで無限に」起動を繰り返されて作動する。

- 5 その基底論理の論理構造は、第3図に示される。

即ち、まず、画面で識別されるメディアに対応する自パレットに属す自単語によって識別されるデータフィールドにその自単語の有意性が成立していないか(データコードが存在していないか)否かを確認する(ステップ301)。

- 10 次に、ステップ301でデータフィールドに自単語の有意性が成立していない(データコードが存在していない)ときには、他のメディアに属し自単語と同じ識別子で指標される自パレットに属すデータフィールドにあるデータコードからの代入の試行か、もしくは当該自単語と同じとは限らない識別子で指標される自パレットに属すデータフィールドにあるデータコードによる演算の試行かのいずれか(この代入と演算によるデータコードの化体化を「データ結合」と呼ぶ)により自単語の有意性を化体化する試行を行い、その化体化された有意性を当該画面で識別されるメディアに係る自パレットに属す当該自単語で識別される作業用のデータフィールドに生成する(ステップ302)。

- 15 あるデータコードによる演算の試行かのいずれか(この代入と演算によるデータコードの化体化を「データ結合」と呼ぶ)により自単語の有意性を化体化する試行を行い、その化体化された有意性を当該画面で識別されるメディアに係る自パレットに属す当該自単語で識別される作業用のデータフィールドに生成する(ステップ302)。
- 20 続いてステップ302の化体化試行が成立したか(データコードが存在しているか)否かを確認する(ステップ303)。

ステップ303で化体化試行が成立している場合は化体した自単語の有意性を正統とするか否かを判断する(ステップ304)。

- 25 ステップ304で「正統」の場合には当該作業用データフィールドに存在している自単語の化体化有意性を自単語のデータフィールドへセットし、かつ当該基底論理の不成立状態を示すフラグの成立化を行う(ス

テップ 3 0 5)。

更に、他の基底論理におけるデータ結合のみによる有意性化体化を再起動させるフラグの設定を行う（ステップ 3 0 6）。

5 以上の説明において、パレット連鎖関数における「システム閉塞に至るまで無限に繰り返す」ということと、パレット関数における「その単語の有意性が成立する（再起動フラグがリセットされる）まで無限に起動を繰り返す」ということ、及び、基底論理における「他の基底論理におけるデータ結合のみによる有意性化体化を再起動させるフラグの設定を行う」ということとは、次のような密接な関係をもっている。

10 即ち、基底論理においては自己の単語の有意性が化体化されたことを以って「他の基底論理におけるデータ結合のみによる有意性化体化を再起動させるフラグの設定」を行う。

次に、パレット関数はその再起動フラグが設定されていればそれを初期化したのち基底論理を再起動する。

15 再起動された基底論理では、

- ・ 自己の単語の有意性が化体化されているか、
- ・ 自己の単語の有意性についての化体化の試行が失敗したか、
- ・ 化体化された自己の単語の有意性が正統でないか、

20 のいずれかならば、何の処理もせずに終了するから再起動フラグがセットされることはない。

総ての基底論理がそのような状態になり、次にパレット関数が作動すると再起動フラグが設定されていないので、その W 0 3 パレットを閉塞する。そして次の W 0 4 パレットを経てその単語の有意性がパレット連鎖関数によって画面に送信される。

25 その具象化された単語の有意性を見たユーザがシステム閉塞の指示をすればパレット連鎖関数は自律的に閉塞するが、システム閉塞ではない

指示をすればパレット連鎖関数が具備する論理によって再度W 0 2パレット、W 0 3パレット及びW 0 4パレットが作動する。つまり、システム閉塞ではない指示に従ってその単語の新たな有意性の化体化が行われる。

- 5 このようなシナリオ関数の「単語の有意性化体化」とは、開発を要望された適用業務や適用機能を説明する文章を構成している単語そのものに対して、

- ・スペシファイイング（前述基底論理のステップ3 0 2）、
 - ・バリデーティング（前述基底論理のステップ3 0 3及び3 0 4）、
 - 10 ・ベリファイイング（前述基底論理のステップ3 0 1）、及び、
 - ・テストイング（前述基底論理のステップ3 0 5）
- というアクティビティを、その単語が有意となるまでn回（nには制限がない）実行しているものと見なすことができる。

- 開発要望の文章をきっかけにして要求事象や処理事象やコンフリクト
- 15 事象とその対処方法を説明する文章がスペシファイイング、バリデーティング、ベリファイイング、及びテストイングという技術者の属人的なアクティビティによってスペシフィケーションというドキュメントに纏められ、それに基づいてインプリメントするというこれまでの方法に対し、L y e eでは個別の単語をその基底論理が有する法則に適合するよう
- 20 うに直接的にプログラミング言語で記述して実行すれば、システム内全単語の基底論理の有意性が総て同時的に成立するように働き、結果としてスペシファイイングとバリデーティングとベリファイイングとテストイングをした結果と等価な状態になるのである。

- 以上の特徴を有するシナリオ関数の持つ仕組みは、実はわれわれ人類
- 25 のメンタルな「思考の仕組み」つまり「意識と認識の仕組み（構造）」即ち「ニューロンの仕組み」そのものとみなせることができる。

第4図は、本発明の一実施形態に係る単語が集合化された文章という開発要望と処理装置との関係を説明する概念図である。

すなわち、意識や認識を成立させる究極的不可分の要素が言動化した事柄をL y e eでは「単語」と呼ぶ。換言すれば「単語」とは「意味」
5 を成立させ、それに基づく生命作用と生命現象を成立させる究極的不可分な要素である。

われわれ人類の言動という生命現象は、メンタルな世界における生命作用が単語ごとの有意性を知覚し、その結果として単語の発言という言動として具象化し、本人及び他人によって認識が可能となったものである
10 考える。

それが例えば適用業務や適用機能をソフトウェアとして実現したいとする開発要望を構成する単語の発言なのであり、またその適用業務や適用機能を例えばスタンドアローン型ソフトウェアに、また例えばクライアント／サーバ型ソフトウェアに、また例えばクライアント／フロント
15 サーバ／アプリケーションサーバ型ソフトウェアになど、処理装置のさまざまな構成に対してインプリメントしたいとする開発要望を構成する単語の発言なのである。

そのようなわれわれ人類の言動という生命現象を引き起こすメンタルな世界の生命作用の仕組み（構造）が持つ普遍的法則を確立したものが
20 取りも直さずシナリオ関数なのである。

つまり、われわれ人類は意識する／しないに関わらず、具象化されている事柄を第6感（気配）で知覚し、視覚、聴覚、触覚、嗅覚、及び味覚などにより生命作用のきっかけを与えられる。そして、その生命作用（即ち、思考作用）の帰結としてその事柄の意味を捉えて認識が行われ、
25 言動と記憶という生命現象を引き起こす。

その具象化された事柄という現象を、さらに、意識する／しないに関

わらず第6感（気配）で知覚し、視覚、聴覚、触覚、嗅覚、及び味覚などにより生命作用のきっかけを与えられる、という具合に生命現象と生命作用とが生命ある限り連綿と繰り返されることになる。

インプリメントするソフトウェアのスペシファイイング、バリデーテ
5 イング、ペリファイイング及びテストイングというアクティビティは、
このような生命現象と生命作用の永遠的な繰り返しによって適用業務や
適用機能を説明するための単語の有意性を成立させようとしていることに
他ならないのである。

次に、このアクティビティをシナリオ関数の仕組みで説明する。

10 まず、開発要望が開発要望者の言動という生命現象として自然言語で
表明する。この開発要望は単語が集合化された文章というものである。

第4図に示すように、この表明に至る開発要望者の生命作用の仕組み
（ニューロンの仕組み）は、シナリオ関数のW04パレットの作用と同
じ仕組みで単語に属す性質（属性）並びに実体（単語の字面）として編
15 集され、前述したパレット連鎖関数のステップ106の作用（画面への
送信）と同じ仕組みで言動化（画面への単語の実体と実体に属す性質で
ある属性とで表示化）されたものと見なすのである。

この開発要望は開発要望者の開発意図が実体とその属性として具現化
したものであるが、その開発意図は、既に具象化されている適用業務や
20 適用機能が呈している事柄を開発要望者の第6感（気配）で知覚し、視
覚、聴覚、触覚、嗅覚、及び、味覚などにより与えられて働いた生命作
用（認識作用）のなせる技であったものとする。

つまり、既に具象化された事柄を開発要望者の第6感（気配）で知覚
し、視覚、聴覚、触覚、嗅覚、及び、味覚などにより与えられるという
25 ことは前述のパレット連鎖関数のステップ108の作用（画面からの受
信）と同じ仕組みで行われたものと同義であるとする。

この画面からの受信を契機に、W 0 2 パレット及びW 0 3 パレットの作用と同じ仕組みによって発言（単語の実体と属性として画面に表示）された開発要望が意図を満たしているかのスペシファイイング、バリデーティング、ベリファイイング及びテストイングをしていくのである。

- 5 従って、開発要望の意図とは既に具象化されている適用業務や適用機能が呈している事柄を説明する単語の実体と属性とを導出する生命作用ということになる。

- 画面表示の形で開発要望の文章を構成する単語の実体と属性が表明されるに至るW 0 2 パレット、W 0 3 パレット及びW 0 4 パレットの作用
10 同じ仕組みの生命作用を可逆化（発言された単語の実体と属性を意識の世界に在った筈のその単語の由来である意識連鎖に位相させ、そこから再度W 0 2 パレット、W 0 3 パレット及びW 0 4 パレットを辿ること）すれば開発要望の意図は満たせるということになる。

- 即ち、表明された開発要望の単語に係るW 0 2 パレットの作用と同じ
15 仕組みのアクティビティを実行し、次にW 0 3 パレットの作用と同じ仕組みのアクティビティを実行し、最後にW 0 4 パレットの作用と同じ仕組みのアクティビティを実行すれば、単語の有意性が化体化され、その化体化された有意性が開発要望の意図として把握できることになるのである。

- 20 結局、シナリオ関数はインプリメントするソフトウェアのスペシファイイング、バリデーティング、ベリファイイング及びテストイングというアクティビティを単語の有意性の化体化という形態により実現していることになるのである。

- これに対して、L y e e 以外の従来方法ではそのようなアクティビテ
25 イを何ら法則性を持たない属人的な経験と知識と能力による生命作用と生命現象をその都度実行しなければならないのである。

L y e eでは、処理装置のさまざまな構成に対して適用業務や適用機能をソフトウェアとしてインプリメントするに際し、適用業務や適用機能からのさまざまな事象と処理装置のさまざまな構成から誘導されるさまざまな事象とのさまざまなコンフリクト（衝突）事象に耐える仕組みを確立するというのも、スペシファイイング、バリデーティング、ベリファイイング及びテストイングという生命作用と生命現象であるのだから、それらはすべてシナリオ関数の仕組み、つまり、適用業務や適用機能並びに処理装置のさまざまな事象を説明する単語を基底論理の仕組みを前提としてプログラミング言語でインプリメントし、それらを集めたW 0 4パレット、W 0 2パレット及びW 0 3パレットをパレット連鎖関数の作用と同じ仕組みで作動させるだけで、即、処理装置のさまざまな構成に適用される適用業務や適用機能のソフトウェアを自動的にスペシファイイング、バリデーティング、テストイング及びベリファイイング及びテストイングというアクティビティが継続的に行われることになるのである。

発明を実施するための最良の形態

以下、図面を参照して本発明の実施形態を説明する。

（１）スタンドアローン型処理装置へソフトウェアをインプリメントするに必要なスペシファイイング、バリデーティング、ベリフィケーション及びテストイング

開発要望者による開発要望は表面的には網羅性と正鵠性に欠けた文章で表明される。例えば「顧客が預金残高を確認できるソフトウェアを開発して欲しい」という具合に、例えば「預金元帳」という言葉が欠如した状態で表明される。

この開発要望は實際上、開発要望者の生命作用（思考作用）で決定さ

20

れ、言動という生命現象として表明されたものであるが、その生命作用と生命現象は開発要望者の深層心理に潜在しているシナリオ関数の持つ法則と等価な法則の成せる技であると本発明では考える。

すなわち、シナリオ関数の各コンポーネントの持つ論理構造はプログラミング言語で記述され、第4図に示したように処理装置の内部（主メモリ）にローディングされて作動する。

第5図は、本発明の一実施形態に係るシナリオ関数の三つめのコンポーネントである単語ごとの基底論理が生命作用を司るニューロンとみなした場合の生命作用と生命現象との関係を説明するブロック図である。

10 処理装置を生命体と見なすと、シナリオ関数の三つめのコンポーネントである単語ごとの基底論理は同図に示すように生命作用50を司るニューロンの如くその生命体の深層心理に内在し、その作用によって単語の発言という言動（生命現象52）が行われたと考えるのである。

すなわち、第5図において、

- 15 ・ L 2 () は W 0 2 パレット 5 0 1 に実装する基底論理（論理要素）
- ・ Y 3 () は W 0 3 パレット 5 0 3 に実装する基底論理（位相要素）
- ・ L 3 () は W 0 3 パレット 5 0 3 に実装する基底論理（論理要素）
- ・ Y 4 () は W 0 4 パレット 5 0 5 に実装する基底論理（位相要素）
- ・ L 4 () は W 0 4 パレット 5 0 5 に実装する基底論理（論理要素）

20 をそれぞれ意味し、ニューロンと見なせるものとする。

括弧内はそのニューロンによる生命作用で決定され生命現象52として表明される開発要望を構成する単語である。

生命現象52として表明されるこの開発要望を満たすソフトウェアをインプリメントするためのスペシファイイング、バリデーティング、ベ
25 リファイイング及びテストイングとは、既に表明された単語ごとのこれらの基底論理がシナリオ関数の有する法則を忠実に満たしてインプリメ

ントされ、作動することを以て自ずから達成される。

従って、開発要望を満たすソフトウェアをインプリメントするための
スペシファイイング、バリデーティング、ベリファイイング及びテスト
5 イングとはシナリオ関数の有する法則を忠実に満たしてインプリメント
するという事と同義となる。

換言すれば、W 0 2 パレット 5 0 1 に実装される基底論理の持つシナ
リオ関数の法則は、言動という現象として開発要望を具象されることにな
る個々の単語の字面という実体を成立させるに相応しいその単語に属
す性質（属性）という有意性を決定する作用である。この作用を達成す
10 るためのシナリオ関数の法則を満たすには、まず、単語が表明されたとい
う事実に着目する。この事実はその単語に対応するニューロンが深層
心理に存在し作用したから生じた筈である。そのニューロンの仕組みを
ソフトウェアとして実現するためのアクティビティはベリファイイング
（その仕様記述の間違いの排除）である。

15 すなわち、その単語に対応するデータ・フィールド並びに基底論理が
実現されているかをチェックするというアクティビティがなされて、問
題のないことが保証されなければならない。

これに対するシナリオ関数の法則は、パレット連鎖関数による W 0 2
パレット 5 0 1 の起動、起動された W 0 2 パレット 5 0 1 のパレット関
20 数による基底論理の起動部分及び基底論理のステップ 3 0 1 部分が該当
する。

パレット連鎖関数による W 0 2 パレット 5 0 1 の起動部分（ステップ
1 0 9）の法則を Visual Basic（以下、これを「V B」と表記する）な
るプログラミング言語で記述すると、

25 *****

Rem***W02パレット起動 Sub を CALL します。

Call W02PALLET

Public Sub W02PALLET()

Rem***

5 Select Case NextPltID

\$LOOP 定義体(画面)

 Case PltID_kW02

 Call kW02

\$ENDLOOP // end of 定義体(画面)

10 End Select

 End Sub

となる。ここで、kはその開発要望として表明された文章を識別する識別子であり、情報処理装置の場合はその開発要望の単語を表示する画面の識別子である。今、前述した開発要望の文章を識別する識別子を「残高照会」とすると、パレット連鎖関数によるパレットの起動部分は各画面ごとに、

15 Case PltID_残高照会 W02

 Call 残高照会 W02

となる。

20 このプログラムの意味するところは、開発要望として表明された文章が画面を介して表明されたとみなした場合、その画面に対応するパレットを引数なしで無条件に起動するということである。この記述によって、対応する基底論理が実装されているパレットが存在しなければコンパイラにより自ずからベリファイイングされ、その仕様記述が間違っただけで動作することがなくなる。即ち仕様記述の間違いの排除がなされることになる。

次に、パレット関数による基底論理の起動部分の法則をVBで記述すると、 Call L2_k_I

となる。ここで、i はやはり開発要望として表明された単語の実体（字面）であり、k もやはりその開発要望として表明された文章を識別する

- 5 識別子である。今、前述した開発要望の文章を識別する識別子を「残高照会」とすると、パレット連鎖関数によるパレットの起動部分は各画面ごとに、

Call L2_残高照会_顧客

Call L2_残高照会_が

- 10 Call L2_残高照会_預金

Call L2_残高照会_残高

Call L2_残高照会_を

Call L2_残高照会_確認

Call L2_残高照会_できる

- 15 となる。

このプログラムの意味するところは、開発要望として表明された文章に存在する単語ごとの基底論理なる関数を引数なしで無条件に起動するということである。この記述によって、対応する基底論理が存在しなければコンパイラにより自ずからベリファイイングされ、その仕様記述が

20 間違ったまま作動することがなくなる。即ち仕様記述の間違いの排除がなされることになる。

- 次に、基底論理のステップ301の部分の法則は、その単語の実体（字面）を成立させ得る属性という有意性が決定されているかどうか、つまり、その単語の実体で指標されるデータ・フィールドがW02パレット
- 25 501に存在し、そのデータ・フィールドに属性を検査できるデータコードが存在するかどうかを検査する法則である。この法則をVBで記述

すると、

If W02.k.i = “” Then Exit Sub End If

となる。従って、基底論理の第1のステップ部分は各単語ごとに、

If W02.残高照会.顧客 = “” Then Exit Sub End If

5 If W02.残高照会.が = “” Then Exit Sub End If

If W02.残高照会.預金 = “” Then Exit Sub End If

If W02.残高照会.残高 = “” Then Exit Sub End If

If W02.残高照会.を = “” Then Exit Sub End If

If W02.残高照会.確認 = “” Then Exit Sub End If

10 If W02.残高照会.できる = “” Then Exit Sub End If

となる。

このプログラムの意味するところは、開発要望として表明された文章に存在する単語ごとのデータフィールド（W02.残高照会.顧客、W02.残高照会.が、W02.残高照会.預金、W02.残高照会.残高、W02.残高照会.を、W02.残高照会.確認、W02.残高照会.できる）がW02パレット501に存在し、属性を検査できるデータコードが存在しない（= “”）かどうかのチェック（If）ということである。この記述によって、対応するデータフィールドが存在しなければコンパイラにより自ずからベリファイイングされることになる。

20 この三つの部分は開発要望として表明された単語とその単語が属す文章に対応する定義体だけを扱っており、開発要望者の意図を一切歪めてはいない。すなわち、開発要望の意図を完璧（100%）に満たしていることになる。

次に、W02パレット501に実装される基底論理のステップ302
25 の部分は、その単語の実体（字面）で識別されるW02パレット501に属するフィールドを、その単語の字面という実体を成立させるに相應

しい属性を定義するスペシファイイングというアクティビティに相当し、同時にそのとおりの属性として定義されているかどうかのバリデーティングとベリファイイングというアクティビティを行うという法則でもある。この法則をVBで記述すると、

5 If Is 属性(W02.k.i) Then Exit Sub End If

となる。ここで、i はやはり開発要望として表明された単語の実体（字面）であり、k もやはりその開発要望として表明された文章を識別する識別子であり、情報処理装置の場合はその開発要望の単語を表示する画面の識別子である。

10 今、例えば、「顧客」という単語は、番号という「数字属性」で「入力属性」と定義すればステップ302のスペシファイイングが満たされ、バリデーティングとベリファイイングはその単語の実体（字面）で識別されるW02パレット501に属するフィールド(W02.残高照会.顧客)の属性がそのとおりの属性（Numeric）かの検査であるから、

15 If IsNumeric(W02.残高照会.顧客) Then Exit Sub End If

となり、スペシファイイングとバリデーティングとベリファイイングというアクティビティとインプリメントが同時に完了したことになるのである。

最後に、W02パレット501に実装される基底論理のステップ303の部分は、前述のバリデーティングとベリファイイングの結果が偽であった場合、その単語の属性という有意性が成立できなかった旨を後続するパレットが判断できるようにしておくというテストの法則である。

この法則をVBで記述すると、

25 W02.k.i_Non = True

となる。ここで、Non は有意性が成立できなかった旨を表すフラグであ

るということを意味し「不成立フラグ」と呼ぶ。

今、例えば、「顧客」という単語の場合は、

W02.残高照会.顧客_Non = True

となり、テストというアクティビティとインプリメントが同時に
5 完了したことになるのである。

次に、W03パレットに実装される基底論理の持つ法則は、言動とい
う現象で開発要望を具象されることになる個々の単語の字面という実体
を成立させるに相応しいバリュー（値を現すデータコード）を生成する
作用である。この作用は位相要素あるいは論理要素のいずれかにより達
10 成する。

位相要素の第1のステップは、その単語の字面で識別されるW02パ
レットに属するフィールドに属性という有意性が成立しているかどうか、
つまり、スペシファイイングとベリファイイングとバリデーティングが
済んでいるかどうかを検査するというのが法則であって、テスト
15 というアクティビティに相当する。

この法則はその単語の字面で識別されるW02パレットに属するフィ
ールドにデータが存在し、不成立フラグがFalseであることを調べるこ
とで十分である。

この法則をVBで記述すると、

20 If W02.k.i = "" And W02.k.i_Non = True Then Exit Sub End
If

となる。

今、例えば、「顧客」という単語は、

If W02.残高照会.顧客 = "" And W02.残高照会.顧客_Non = True Then
25 Exit Sub End If

となり、テストというアクティビティとインプリメントが同時に

完了したことになるのである。

次の第2のステップは、属性という有意性が成立していればそのデータコードをその単語の字面で識別されるW03パレット503に属するフィールドにセットするというスペシファイイングの法則である。この

5 法則をVBで記述すると、

W03.k.i = W02.k.i

となる。

今、例えば、「顧客」という単語は、

W03.残高照会.顧客 = W02.残高照会.顧客

10 となり、スペシファイイングというアクティビティとインプリメントが同時に完了したことになるのである。

論理要素の第1のステップは、その単語の字面で識別されるW03パレットに属するフィールドにバリューが生成されていないかどうか、つまり、位相要素によってスペシファイイングが完了しているかどうかの

15 テスティングの実行という法則である。この法則をVBで記述すると、

If W03.k.i <> "" Then Exit Sub End If

となる。

今、例えば、「顧客」という単語は、

If W03.残高照会.顧客 <> "" Then Exit Sub End If

20 となり、テストというアクティビティとインプリメントが同時に完了したことになるのである。

次の第2のステップは、その単語の字面で識別されるW03パレット503に属するフィールドにバリューが生成されていなければ、その単語のバリューをその単語の字面で識別されるW03パレット503に属

25 する作業用フィールドに生成するという法則であり、スペシフィケーションというアクティビティに相当する。例えば、

・「顧客」という単語は「入力属性」であるから位相要素で生成されるので論理要素としてはロジックを持たない。

・「が」という単語は単なる「ラベル属性」であるからロジックを持たない。

5 ・「預金」という単語は「入力属性」であるから位相要素で生成されるので論理要素としてはロジックを持たない。

・「残高」という単語は「出力属性」であるから元帳に属する残高で生成されるので論理要素としてはロジックを持たない。

10 ・「を」という単語は単なる「ラベル属性」であるからロジックを持たない。

・「確認」という単語は「入力属性」であるから位相要素で生成されるので論理要素としてはロジックを持たない。

・「できる」という単語は単なる「ラベル属性」であるからロジックを持たない。としてスペシファイイングする。この法則をVBで記述すると、

15 W03_WORK.k.i = バリユー生成式

となる。

今、例えば、「残高」という単語は、「元帳に属する残高で生成」としてスペシファイイングされたので、

W03_WORK.残高照会.残高 = W03.元帳.残高

20 となり、スペシファイイングというアクティビティとインプリメントが同時に完了したことになるのである。

ここで、「元帳に属する残高」という当初の開発要望としては顕在化しておらず当初の開発要望に潜在していた単語がスペシファイイングの法則を満たすようにアクティビティすることによって浮かび上がってきたことになる。この「元帳に属する残高」という新たな単語が及ぼす影響とシナリオ関数の法則との関係については後述する。

25

次の第3のステップは、前述の第2のステップが成立したか（バリューが生成できたか）否かを確かめるベリファイイングに相当するアクティビティの法則である。この法則をVBで記述すると、

If W03_WORK.k.i = “” Then Exit Sub End If

5 となる。

今、例えば、「残高」という単語は、

If W03_WORK.残高照会.残高 = “” Then Exit Sub End If

10 となる。何故に、これがベリファイイングかといえ、前述の第2のステップが成立するためにはバリュー生成式が正しくなければならないからである。

例えば、

W03_WORK.残高照会.残高 = W03.元帳.残高

15 というバリュー生成式が必然的に成立するためには、右辺のデータフィールドが存在していることが必要であり、その為には右辺に対応する単語が定義されていなければならないのである。

しかし、この時点では「元帳上の残高」という単語は第5図上に現れていない。

20 この結果、このプログラムはこのままでは必ず異常終了することになる。この異常終了を引き起こさないようにすることこそベリファイイングの目的であるから、この第3ステップの法則を満たしたことにより、このままでは不足する「元帳上の残高」という単語をこの開発要望を説明する単語として充足するというアクティビティが必然的に必要になったことになる。

25 結果として、ベリフィケーションというアクティビティとインプリメントが同時に完了したことになるのである。そして「元帳上の残高」という単語はバリューという有意性が記憶ということを司る元帳という媒

30

体に属する単語であるから、シナリオ関数の法則によればシナリオ関数
T 1 で扱う単語となり、かつ、境界単語としてシナリオ関数 T 0 にも属
させることがシナリオ関数の法則から導き出せるのである。このように
して、バリューという有意性を生成させるスペシフィケーションをシナ
5 リオ関数の法則を満たすように執行していけば完璧なスペシフィケーシ
ョンが自ずから達成されるのである。

次の第 4 のステップは、前述の第 3 のステップが成立している場合は
化体したバリューという有意性を正統とするか否かを判断するバリデー
ティングに相当するアクティビティの法則である。この法則は前述の第
10 2 のステップのバリュー生成式が複数の単語で演算する場合のみ必要と
なる法則である。

すなわち、前述のようにバリュー生成式が 1 個の単語による代入式の
場合は前述の第 3 のステップでペリファイニングされ、その結果として
バリデーティングもされるのであるが、複数の場合は譬えペリファイイ
15 ングされていたとしても、数学的な意味で生成が成立するとは限らない
からである。例えば、 $A = B + C$ という計算式が必然的に成立するため
には、B と C のいずれかに有意なバリューが存在しなければならないの
である。

このような理由から、この第 4 のステップでは前述の第 2 のステップ
20 のバリュー生成式が複数の単語で演算する場合のみ、その生成式を必然
的に成立させる条件を調べるというバリデーティングに相当するアクテ
ィビティを行う。

例えば、前記「 $A = B + C$ 」の場合は、

```
If W03.残高照会.B = “” And W03.残高照会.C = “” Then  
25 Exit Sub End If
```

となり、バリデーティングというアクティビティとインプリメントが同

時に完了したことになるのである。

次の第5のステップと第6のステップは、前述の第4のステップが正統の場合には前述の作業用データフィールドに生成されているバリューという有意性を自己のデータフィールドへセットし、かつ当該基底論理

5 の不成立状態を示すフラグの成立化を行い、更にW03パレット503に属する他の総ての基底論理におけるデータ結合のみによるバリューという有意性化体化を再起動させるフラグの設定を行うというバリデーションに相当するアクティビティの法則である。この法則をVBで記述すると、

10 W03.k.i = W03_WORK.k.i
 W03.k.i_Non = False
 W03.Recall_FLG = True

となる。ここで、Nonは有意性が成立できなかった旨を表す「不成立フラグ」であることを意味し、W03.Recall_FLGは再起動させるフラグで

15 「再起フラグ」と呼ぶ。

今、例えば、「残高」という単語は、

W03.残高照会.残高 = W03_WORK.残高照会.残高
W03.残高照会.残高_Non = False
W03.Recall_FLG = True

20 となり、バリデーションというアクティビティとインプリメントが同時に完了したことになるのである。

このようにこの単語にかかるバリューという有意性が成立して再起フラグがセットされた結果、W03パレットに実装されたすべての基底論理が再び作動し始めることになる。そして、第1のステップでバリュー

25 という有意性が成立しているか、第3のステップでバリューという有意性が成立していないか、第4のステップでバリューという有意性が正統

でないかのいずれかの状態にすべての基底論理が同時に至るまで無限に繰り返されることになる。

即ち、無限回のスペシファイイングとバリデーティングとベリファイイングとテストイングがされ、途中、異常終了することなしに、すべての
5 の単語において第1のステップでバリューという有意性が成立しているか、第3のステップでバリューという有意性が成立していないか、第4のステップでバリューという有意性が正統でないかのいずれかの状態に同時に至れば不定でない完璧なソフトウェアが生成されたことになる。

上述した一連の働きを、同期式スペシファイイング、同期式ベリファイ
10 イング、同期式バリデーティング及び同期式テストイングと呼ぶ。なお、第3のステップでバリューという有意性が成立していない場合と、第4のステップでバリューという有意性が正統でない場合は、インプリメントされたソフトウェアを利用する人間側の操作事象か処理装置の故障事象かに限定されるので次のW04パレットに実装される基底論理に
15 よって利用者への報告事象として実現されることになる。

次に、W04パレット505に実装される基底論理の持つ法則は、言動という現象で開発要望を具象されることになる個々の単語の字面という実体を成立させるに相応しい「形式」という属性にかかる有意性を編集する作用である。この作用は位相要素及び論理要素とにより達成する。

20 位相要素の第1のステップは、その単語の字面で識別されるW02パレット501に属するフィールドに属性という有意性が成立しているかどうかを調べ、もし成立していれば、それをその単語の字面で識別されるW04パレット505に属するフィールドにセットする。もし、成立していなければ、その単語の字面で識別されるW03パレット503に
25 属するフィールドに属性という有意性が成立しているかどうかを調べ、もし、成立していればそれをその単語の字面で識別されるW04パレッ

ト 5 0 5 に属するフィールドにセットするというテストとバリデーションとスペシフィケーションというアクティビティに相当するのが法則である。この法則を V B で記述すると、

If W02.k.i <> "" And W02.k.i_Non = False Then

5 W04.k.i = W02.k.i

Else

 If W03.k.i <> "" Then

 W04.k.i = W03.k.i

 End If

10 End If

となる。

今、例えば、「残高」という単語は、

If W02.残高照会.残高 <> "" And W02.残高照会.残高_Non = False
Then

15 W04.残高照会.残高 = W02.残高照会.残高

Else

 If W03.残高照会.残高 <> "" Then

 W04.残高照会.残高 = W03.残高照会.残高

 End If

20 End If

となり、テストというアクティビティとインプリメントが同時に完了したことになるのである。

論理要素の第 1 のステップは、その単語の字面で識別される W 0 4 パ
レットに属するフィールドにバリューという有意性が成立しているかど
25 うか、つまり、位相要素によってバリューという有意性が成立している
かかどうかを検査するバリデーションというアクティビティに相当する

法則である。しかし、W 0 2 パレット 5 0 1 及び W 0 3 パレット 5 0 3 を異常終了することなしに作動したということは、ここまでのスペシファイイングとベリファイイングとバリデーティングとテストイングは完璧であることを意味するのであるから、何らかの有意性は必ず成立していることになる。従って、この第 1 のステップの結果は必ず真となり第 2 のステップに進めるようであればならない。

この法則を V B で記述すると、

If W04.k.i <> W04.k.i Then

Exit

10 End If

となる。

今、例えば、「残高」という単語の場合は、

If W04.残高照会.残高 <> W04.残高照会.残高 Then

Exit Sub

15 End If

となる。

次の第 2 のステップは、そのバリューという有意性をその単語の字面で識別される W 0 4 パレット 5 0 5 に属するフィールドにその単語の字面という実体を成立させるに相応しい形式に編集するというスペシフィケーションというアクティビティに相当する法則である。その単語の字面という実体を成立させるに相応しい形式に編集するということは、例えば、「残高」という単語は「通貨記号を数字の先頭に付与する」という形式に編集する、というスペシファイイングに相当する。これを V B で記述すると、

25 W04.残高照会.残高 = “¥” + W04.残高照会.残高

となり、スペシファイイングとインプリメントとが同時に完了する。

35

次の第3のステップは、前述のW02パレット501乃至W03パレット503の基底論理が目的を達成したかどうかを否かを確かめ、不成立フラグがセットされていたならば、不成立の旨をMSGなる単語で報告するスペシファイイングというアクティビティに相当する法則である。

5 この法則をVBで記述すると、

```
      If W02.k.i_Non = True Then
          MSG = "W02.k.i error"
      Else
          If W03.k.i_Non = True Then
10         MSG = "W03.k.i error"
          End If
      End If
```

となる。

今、例えば、「残高」という単語の場合は、

```
15      If W02.残高照会.残高_Non = True Then
          W04.残高照会.MSG = "W02.残高照会.残高 error"
      Else
          If W03.残高照会.残高_Non = True Then
              W04.残高照会.MSG = "W03.残高照会.残高 error"
20          End If
      End If
```

となり、エラー事象に対するスペシファイイングとインプリメントとが同時に完了するのである。

このようにして、開発要望を表明するために発言された単語からその
25 開発要望を満たすソフトウェアをインプリメントするためのスペシフィケーションとバリデーションとベリフィケーションというドキュメント

を作成することなく、その単語の発言に至るメンタルな領域での思考の仕組みをシナリオ関数の法則に当てはめてプログラミング言語でインプリメントするだけで、スペシファイイングとバリデーティングとベリファイイングとテストイングというアクティビティが執行されたと等価なソフトウェアを完成できるのである。

第6図は、このようにしてインプリメントされたプログラムの構成と作動の順序を説明するブロック図である。

同図において二重矢線はパレット連鎖関数 $\Phi 0(60)$ というプログラムである。 $\Phi p4(601)$ 、 $\Phi p2(603)$ 及び $\Phi p3(605)$ はパレット関数である。 $Yn()$ は位相要素であり $Ln()$ は論理要素である。作用要素は図中省略してある。

さて、開発要望の表明は画面表示と等価である。そして、画面上の単語はその単語ごとの5つの基底論理が有する法則と等価とみなせるニューロンの作用でその実体と属性という有意性が決定され発言された事柄と考える。

その基底論理が有する法則は前述のとおりその単語の有意性決定のためのスペシファイイングとバリデーティングとベリファイイングとテストイングの法則を具現化したのであるから、このような基底論理が作動するということは開発要望として表明された単語のスペシファイイングとバリデーティングとベリファイイングとテストイングというアクティビティが同時に行われているといえることができるのである。

(2) クライアント／サーバー型処理装置へソフトウェアをインプリメントするに必要なスペシファイイング、バリデーティング、ベリフィケーション及びテストイング

前述したように、ソフトウェアを開発したいという要望は自然言語の単語からなる文章で表明される。その文章を構成する一つの単語の表明

に至るメンタルな領域での思考の仕組みをシナリス関数の仕組みと等価であると見なせば、その単語はその単語の実体に属す性質（属性）との関係だけでバリューがセットされ、その結果として表明という事実に至ったのではなく、一つの単語のバリューがセットされたその延長でこの

5 単語にかかる再起フラグがセットされ、その帰結としてその単語にかかる元帳上の残高を含むW 0 3パレットに属しているすべての単語の基底論理が同期式にスペシファイイングされ、同期式にベリファイイングされ、同期式にバリデーションされかつ同期式にテストされた結果であるということになる。

- 10 即ち、スタンドアローン型に対して適用業務や適用機能をソフトウェアとしてインプリメントするということは、開発要望者がメンタルな領域でその適用業務や適用機能をイメージしていたのである。その開発要望者という生命体を一つの処理装置とし、その思考作用はその一つの処理装置上の主メモリにストアしたシナリオ関数で規定されたソフトウェアと見なした場合の実現形態がスタンドアローン型なのである。
- 15

この適用業務を例えばクライアント／サーバ型のソフトウェアとしてインプリメントするということは、もともと一つの生命体でイメージしたものを今度は二つの生命体で一つの適用業務或いは適用機能をイメージする場合に相当する。

- 20 本来一つの生命体で執行しようとしていた適用業務あるいは適用機能を二つの生命体で同じ適用業務あるいは適用機能を執行するに際しては、二つの生命体の勝手な要求事象や処理事象とのコンフリクト事象をあたかも一つの生命体として作用するように対処しなければならない。この対処方法がいわゆるプロトコルという概念である。

- 25 第7図は、本発明の一実施形態に係る適用業務や適用機能を二つの処理装置がクライアントとサーバ型とに分かれて執行するソフトウェアを

シナリオ関数の立場で説明する概念的ブロック図である。

同図に示すとおり、一つの処理装置 7 0 上での適用業務や適用機能を二つの処理装置、クライアント 7 2 とサーバ 7 4 とに分かれて執行することをシナリオ関数の法則で説明すれば、

- 5 シナリオ関数 T 0 をクライアントに、
 シナリオ関数 T 1 をサーバに、
それぞれ割り当てるということになる。

- そして、スタンドアローン型 7 0 の場合、シナリオ関数 T 0 とシナリオ関数 T 1 との関係は一つの生命体を前提としていたのであるからプロ
10 トコルは一切不要で双方は W 0 3 パレット (7 0 1、7 0 3) を介して
 逐次的に交信するだけであったのであるが、クライアント／サーバ型 (7
 2、7 4) にあっては二つの生命体はそれぞれ自律的な生命作用と生命
 現象を行うので、ここにコンフリクト事象が発生する。しかし、適用業
 務や適用機能を遂行するという使命は維持されざるを得ないので、双方
15 の生命体同士の間でプロトコルが自ずから行われなければならない。

 第 8 図は、双方の生命体同士の間でひとつの目的を達成するに必要な
 となるプロトコルを説明するためのシーケンシャル・チャートである。

- 同図に示すように、適用業務や適用機能を二つの処理装置でクライ
 アントとサーバとに分かれて分担するソフトウェアとしてインプリメント
20 する場合に従来法では必須とされていたプロトコル制御のための状態遷
 移図やデシジョンテーブルの作成ということをせずに、適用業務や適用
 機能を遂行するために二つの生命体から表明される第 8 図に示されるプ
 ロトコルに係る単語をシナリオ関数 T 1 の法則に当てはめプログラミング
 言語でインプリメントするだけで、従来におけるプロトコルの概念が
25 達成されるのである。

 具体的には第 9 図に示すように、プロトコルに係る単語 (9 0 1、9

03、905、907、909)を、シナリオ関数T1を具現化するプログラムである各パレットW04(92)、W02(94)、W03(96)中の定型的な(変数)識別子挿入位置に貼り付けることで実現される。

- 5 因みにW04(92)、W02(94)、W03(96)は、上述のように、シナリオ関数の持つ法則性により、かかる定型的な(変数)識別子挿入位置以外はプログラムの種類・機能、単語の異同等に関係なく所与の構造を持つ。

- (3) クライアント／フロントサーバ／アプリケーションサーバ型処理装置へソフトウェアをインプリメントするに必要なスペシファイイング、
10 バリデーティング、ベリフィケーション及びテストイング

- クライアント／フロントサーバ／アプリケーションサーバからなるインターネットという処理装置構成の基で一つの適用業務や適用機能を遂行するソフトウェアをインプリメントする場合のスペシファイイング、
15 バリデーティング、ベリフィケーション及びテストイングというアクティビティをシナリオ関数ではどのようなになるかを考察する。

- 第10図は、本発明の一実施形態に係る適用業務や適用機能を三つの処理装置がクライアントとフロントサーバとアプリケーション・サーバとに分かれて執行するソフトウェアをシナリオ関数の立場で説明するための概念的ブロック図である。
20

- インターネットという処理装置構成はクライアント／サーバ型を2層と見ると同図に示すように、一見3層で構成されているかのように見える。しかしながら、実態は、2層からなるクライアント／サーバ(b)におけるクライアント部分110の画面部分110Aと処理装置部分1
25 110Bとを分割した(130、140)にすぎない。これは、クライアント／サーバ型(b)において行っていた画面の定義をHTMLという

インターネット特有の言語でおこない、その画面をブラウザとスクリプトと呼ぶ標準化されたソフトウェアで制御する方式である。

従って、基本的には適用業務や適用機能をクライアント／フロントサーバ／アプリケーションサーバ型（c）の処理装置 140，150 上のソフトウェアとしてインプリメントする場合のスペシファイイング、バリデーティング、ベリフィケーション及びテストティングにおいては、クライアント（画面 130）とフロントサーバ 140 との間を HTML で定義された情報を画面 130 に送信し、URL で定義された画面 130 からの情報を受信するコマンド作用要素の法則に当てはめてインプリメントし、他の部分はクライアント／サーバ型（b）の場合と同じ方法でインプリメントすれば良いのである。

第 11 図は、以上詳細に説明した点を概念的に表したコンセプトチャール・チャートである。

同図に示されるように、開発要望を表明するために発言された単語からその開発要望を満たすソフトウェアをインプリメントするためのスペシフィケーションとバリデーションとベリフィケーションというドキュメントを作成することなく、その単語の発言に至るメンタルな領域での思考の仕組みをシナリオ関数の法則に当てはめてプログラミング言語でインプリメントするだけで、スペシファイイングとバリデーティングとベリファイイングとテストティングというアクティビティが執行されたと等価なソフトウェアを完成できるのである。

以上詳細に説明したように、本実施形態に係る本発明のソフトウェアの同期式スペシファイイング（specifying）方法、バリデーティング（validating）方法、ベリファイイング（verifying）方法、及びテストティング（testing）方法によれば、完璧な製品にするために必要とされるスペシファイイング、バリデーティング、ベリファイイング及びテスト

ングというアクティビティの無限回の繰り返しを実施したと同等の成果を、無限回の繰り返し構造によって単語の有意性を決定する仕組みを内在しているシナリオ関数の法則そのものをプログラミング言語で直接的にインプリメントすることだけで、達成が可能となる。

5

産業上の利用可能性

以上詳述したように、本発明によれば、無限回の繰り返し構造によって単語の有意性を決定する仕組みを内在しているシナリオ関数の法則そのものをプログラミング言語で直接的にインプリメントすることにより、
10 スペシファイイングとバリデーティングとベリファイイングとテストイングというアクティビティが執行されたと等価なソフトウェアを完成できる。

なお、本発明は、上述した実施形態には限定されず、本発明の技術思想の範囲内で様々な変形が可能である。

15 例えば、上述した実施形態では、主にソフトウェアを実現する方法を例にとり説明したが、例えばかかるソフトウェア実現方法をプログラムすることも可能である。

また、上述した実施形態において、定義体は上述したものに限定されず、例えば伝送媒体、紙媒体のようなものであっても本発明は実現され
20 る。

請 求 の 範 囲

1. 生産するシステムに係る画面に表示するための送信情報を決定する第1のステップと、
- 5 前記送信情報を編集するW04パレットの識別子を記録しているデータ・フィールドのデータコードを検査する第2のステップと、
前記第2のステップの結果、画面に表示するための送信情報がメニューである場合には当該システム内の総てのパレットを起動対象としてセットする第3のステップと、
- 10 前記第2のステップの結果、当該画面に表示するための送信情報がメニューでない場合には必要（新規／継続の指定）に応じて使用済みデータ・フィールドを初期化する第4のステップと、
前記送信情報を編集するW04パレットを起動する第5のステップと、
前記第5のステップにおいてW04パレットにより編集されたデータ
- 15 コードを当該画面に送信する第6のステップと、
前記画面上で操作されたデータコードの入力及びその処理指令とを受信する第7のステップと、
前記受信されたデータコード及び処理指令を受け入れるW02パレットを起動する第8のステップと、
- 20 必要に応じてW03パレット或いはT1にかかるパレット連鎖関数を起動する第9のステップと、
前記第1のステップ乃至第9のステップをシステム閉塞に至るまで無限に繰り返す第10のステップと
を具備することを特徴とするソフトウェアの同期式スペシファイング
- 25 グ（specifying）方法。
2. 生産するシステムに係る画面に表示するための送信情報を決定する

第 1 のステップと、

前記送信情報を編集する W 0 4 パレットの識別子を記録しているデータ・フィールドのデータコードを検査する第 2 のステップと、

- 5 前記第 2 のステップの結果、画面に表示するための送信情報がメニューである場合には当該システム内の総てのパレットを起動対象としてセットする第 3 のステップと、

前記第 2 のステップの結果、当該画面に表示するための送信情報がメニューでない場合には必要（新規／継続の指定）に応じて使用済みデータ・フィールドを初期化する第 4 のステップと、

- 10 前記送信情報を編集する W 0 4 パレットを起動する第 5 のステップと、
前記第 5 のステップにおいて W 0 4 パレットにより編集されたデータコードを当該画面に送信する第 6 のステップと、

前記画面上で操作されたデータコードの入力及びその処理指令とを受信する第 7 のステップと、

- 15 前記受信されたデータコード及び処理指令を受け入れる W 0 2 パレットを起動する第 8 のステップと、

必要に応じて W 0 3 パレット或いは T 1 にかかるパレット連鎖関数を起動する第 9 のステップと、

- 20 前記第 1 のステップ乃至第 9 のステップをシステム閉塞に至るまで無限に繰り返す第 1 0 のステップと

を具備することを特徴とするソフトウェアの同期式バリデーティング（validating）方法。

3．生産するシステムに係る画面に表示するための送信情報を決定する第 1 のステップと、

- 25 前記送信情報を編集する W 0 4 パレットの識別子を記録しているデータ・フィールドのデータコードを検査する第 2 のステップと、

前記第 2 のステップの結果、画面に表示するための送信情報がメニューである場合には当該システム内の総てのパレットを起動対象としてセットする第 3 のステップと、

- 5 前記第 2 のステップの結果、当該画面に表示するための送信情報がメニューでない場合には必要（新規／継続の指定）に応じて使用済みデータ・フィールドを初期化する第 4 のステップと、

前記送信情報を編集する W 0 4 パレットを起動する第 5 のステップと、

前記第 5 のステップにおいて W 0 4 パレットにより編集されたデータコードを当該画面に送信する第 6 のステップと、

- 10 前記画面上で操作されたデータコードの入力及びその処理指令とを受信する第 7 のステップと、

前記受信されたデータコード及び処理指令を受け入れる W 0 2 パレットを起動する第 8 のステップと、

- 15 必要に応じて W 0 3 パレット或いは T 1 にかかるパレット連鎖関数を起動する第 9 のステップと、

前記第 1 のステップ乃至第 9 のステップをシステム閉塞に至るまで無限に繰り返す第 1 0 のステップと

を具備することを特徴とするソフトウェアの同期式ベリファイイング（verifying）方法。

- 20 4．生産するシステムに係る画面に表示するための送信情報を決定する第 1 のステップと、

前記送信情報を編集する W 0 4 パレットの識別子を記録しているデータ・フィールドのデータコードを検査する第 2 のステップと、

- 25 前記第 2 のステップの結果、画面に表示するための送信情報がメニューである場合には当該システム内の総てのパレットを起動対象としてセットする第 3 のステップと、

45

前記第 2 のステップの結果、当該画面に表示するための送信情報がメニューでない場合には必要（新規／継続の指定）に応じて使用済みデータ・フィールドを初期化する第 4 のステップと、

前記送信情報を編集する W 0 4 パレットを起動する第 5 のステップと、

- 5 前記第 5 のステップにおいて W 0 4 パレットにより編集されたデータコードを当該画面に送信する第 6 のステップと、

前記画面上で操作されたデータコードの入力及びその処理指令とを受信する第 7 のステップと、

- 10 前記受信されたデータコード及び処理指令を受け入れる W 0 2 パレットを起動する第 8 のステップと、

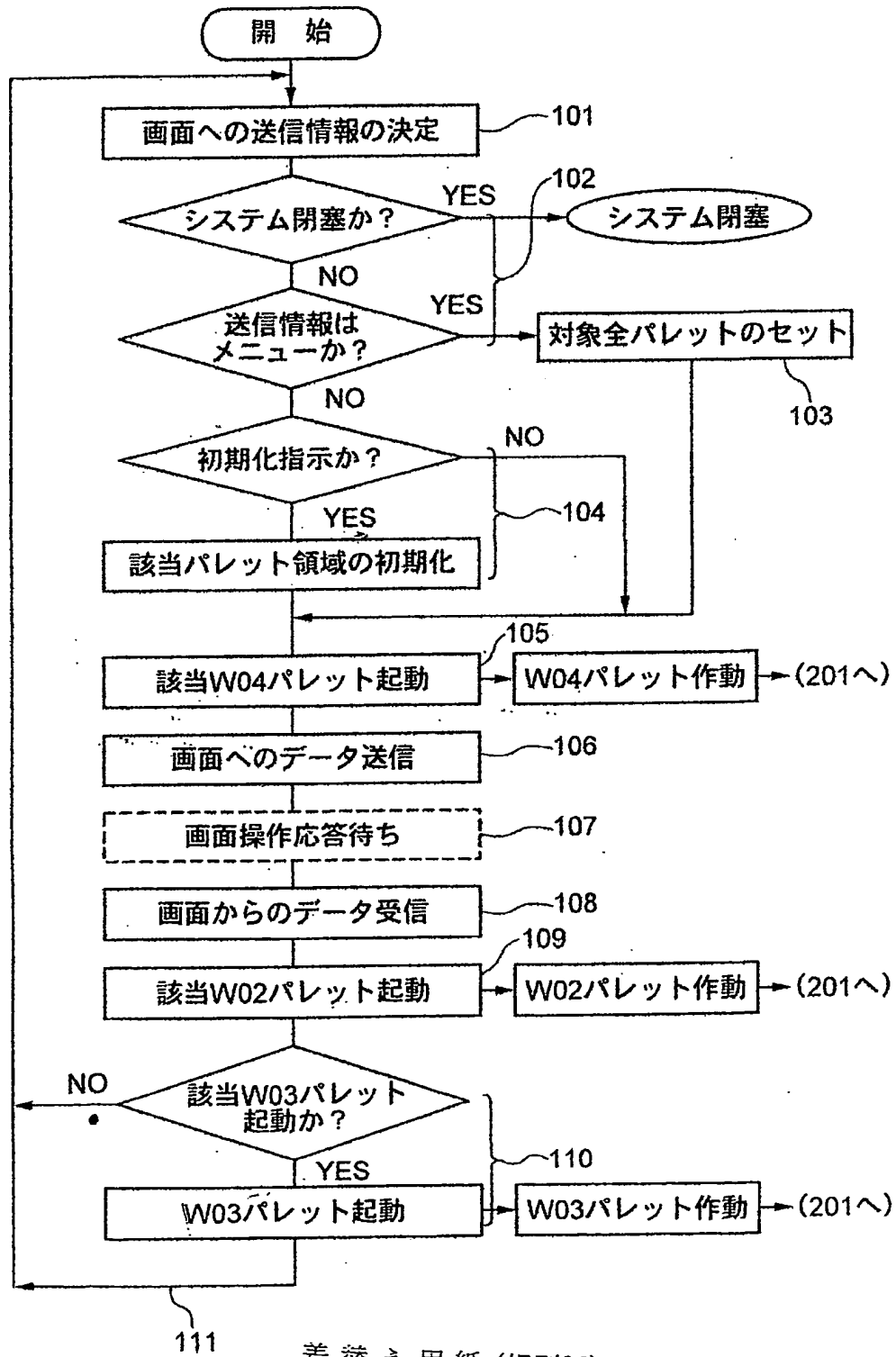
必要に応じて W 0 3 パレット或いは T 1 にかかるパレット連鎖関数を起動する第 9 のステップと、

前記第 1 のステップ乃至第 9 のステップをシステム閉塞に至るまで無限に繰り返す第 1 0 のステップと

- 15 を具備することを特徴とするソフトウェアの同期式テストイング（testing）方法。

1 / 10

図 1



2 / 1 0

図 2

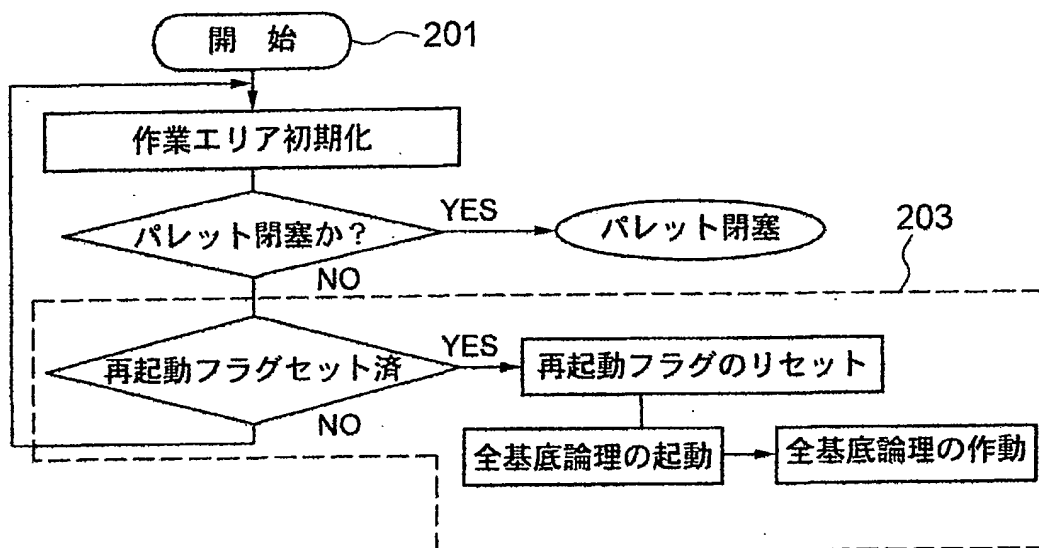
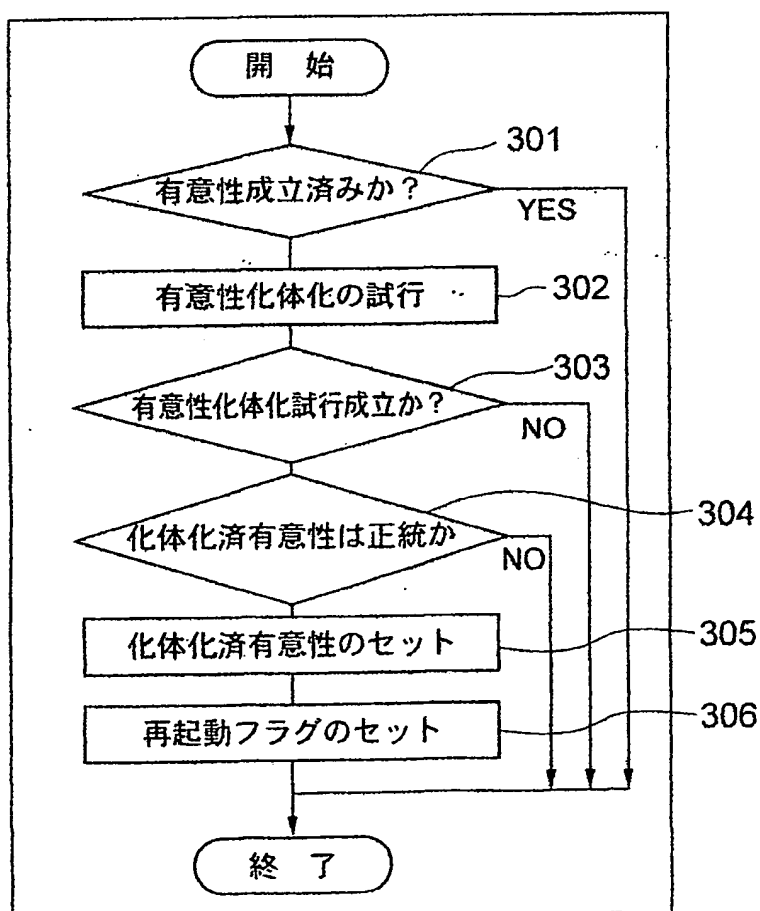
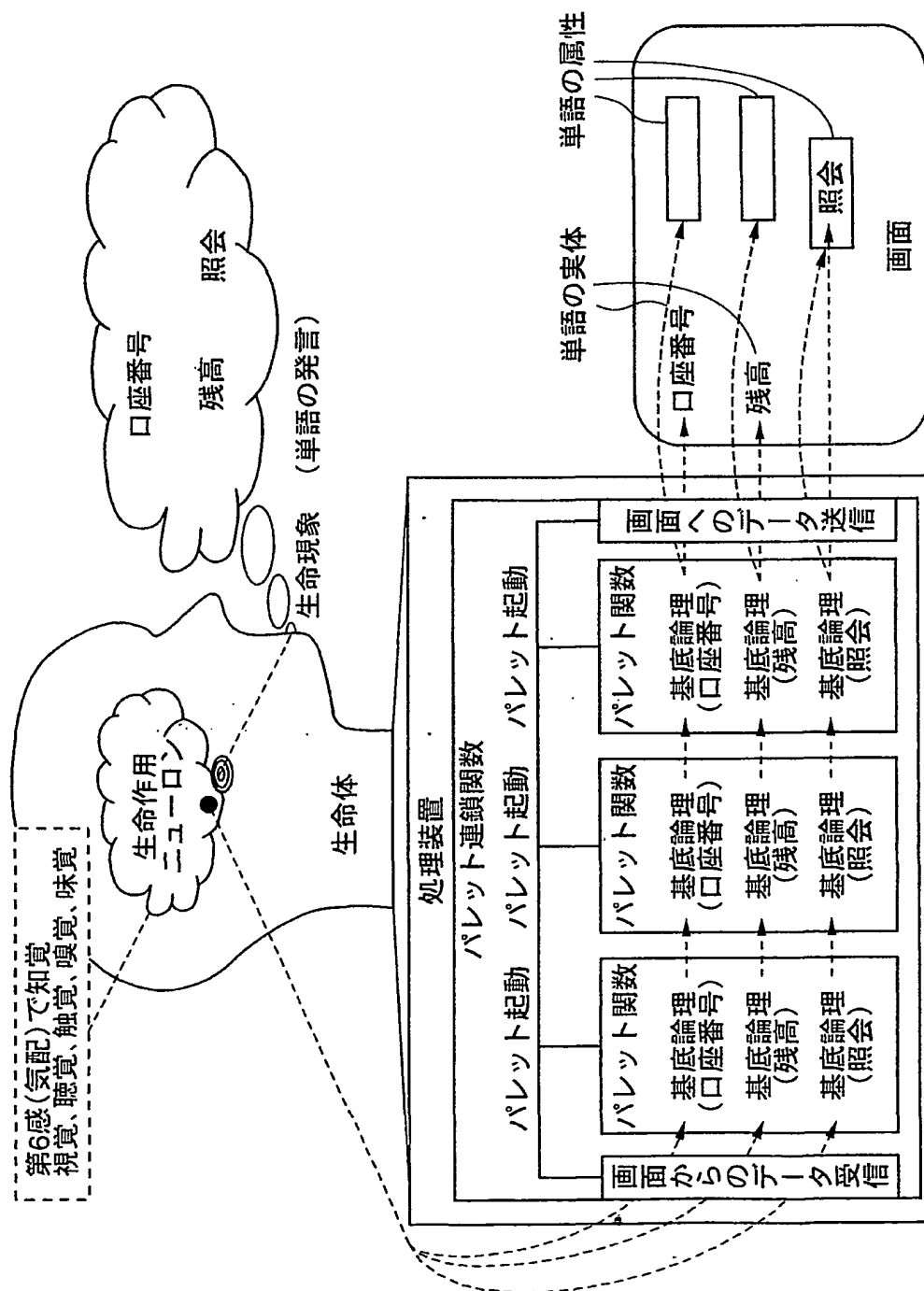


図 3



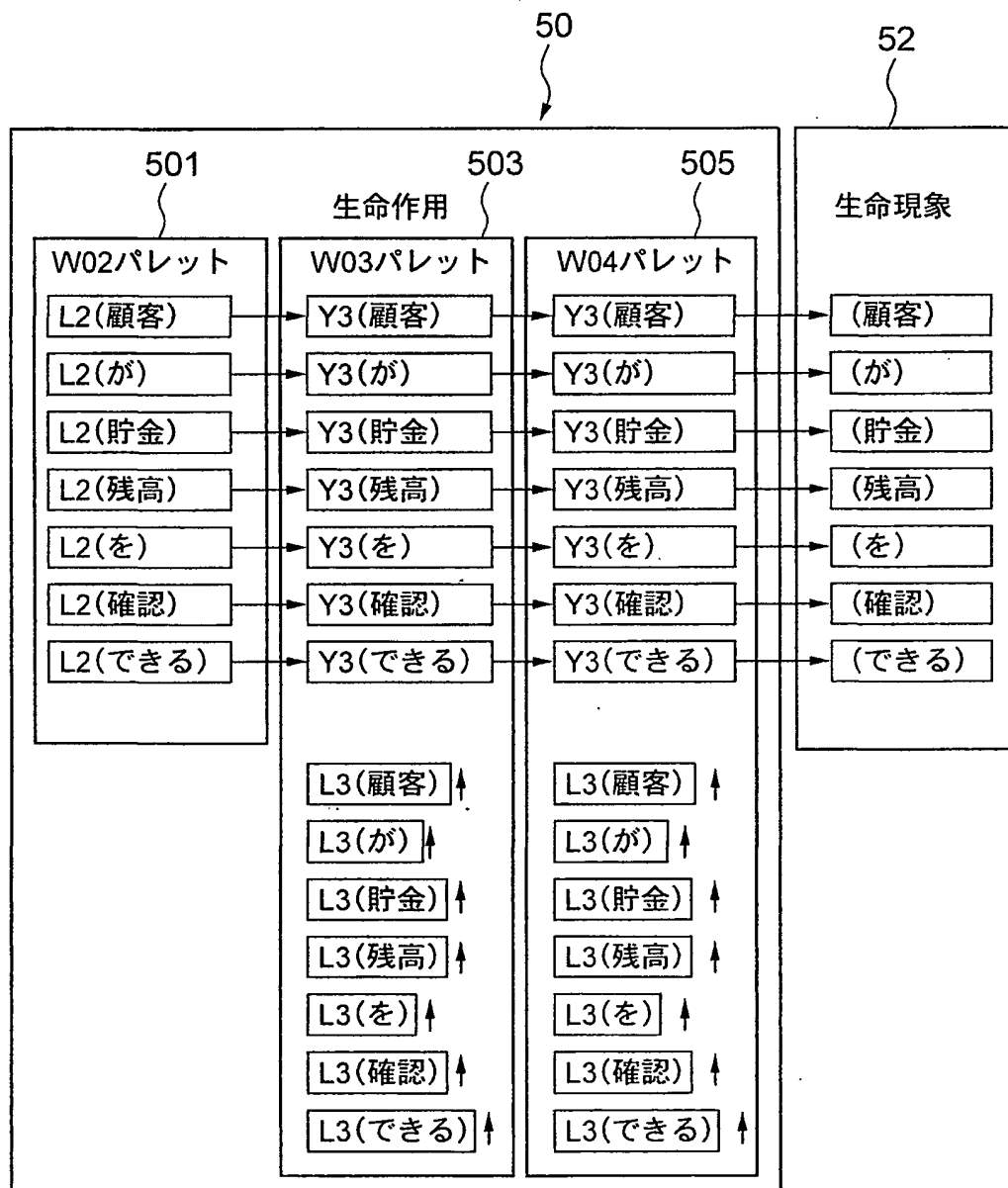
3 / 1 0

図 4



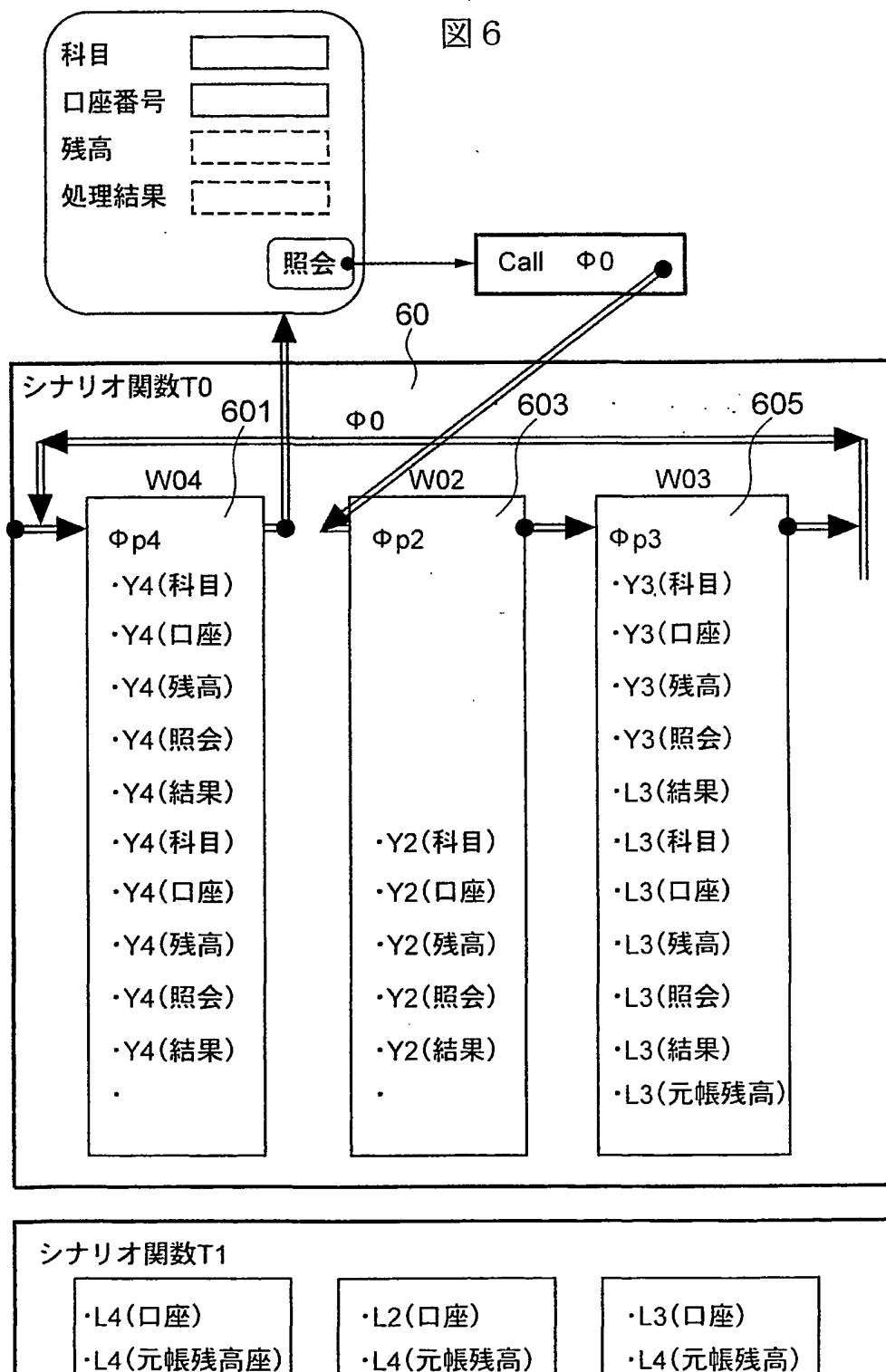
4 / 1 0

図 5



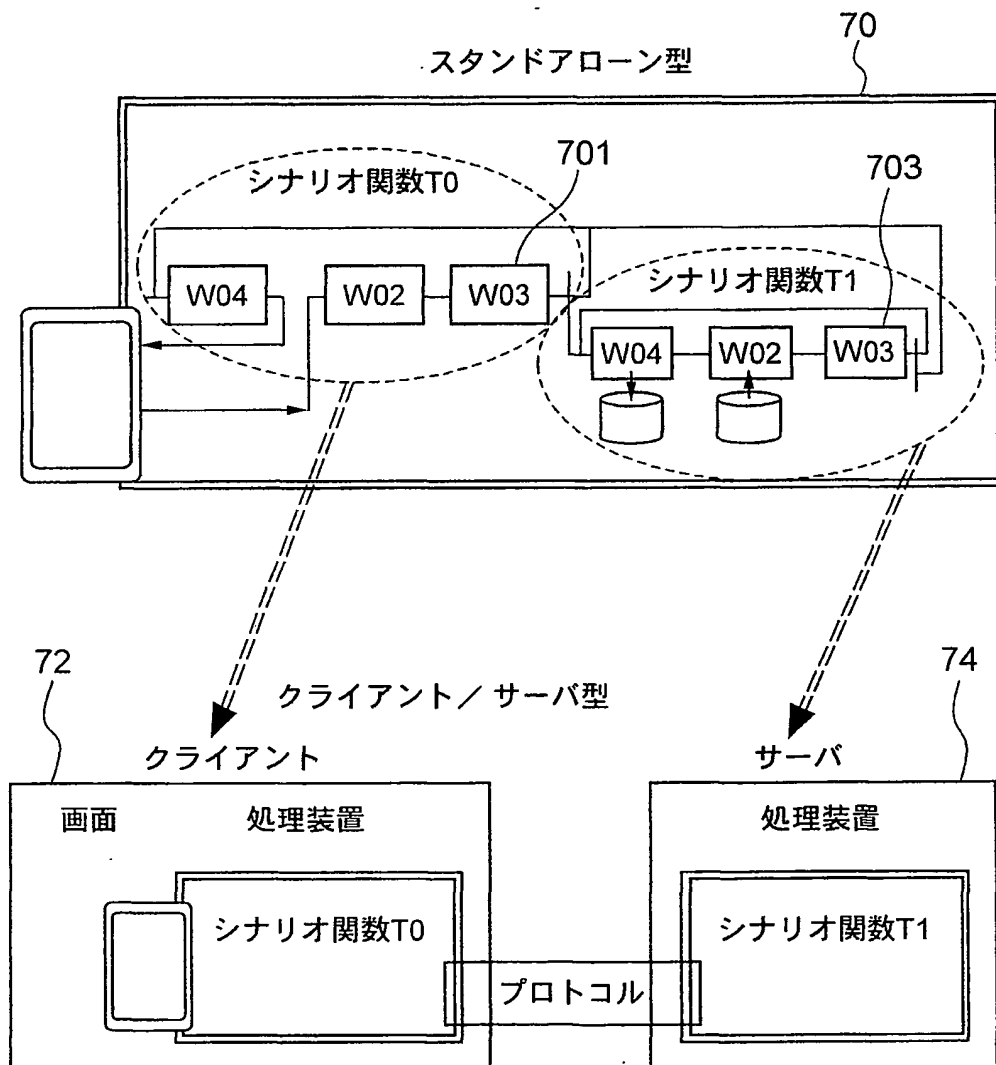
5 / 10

図 6



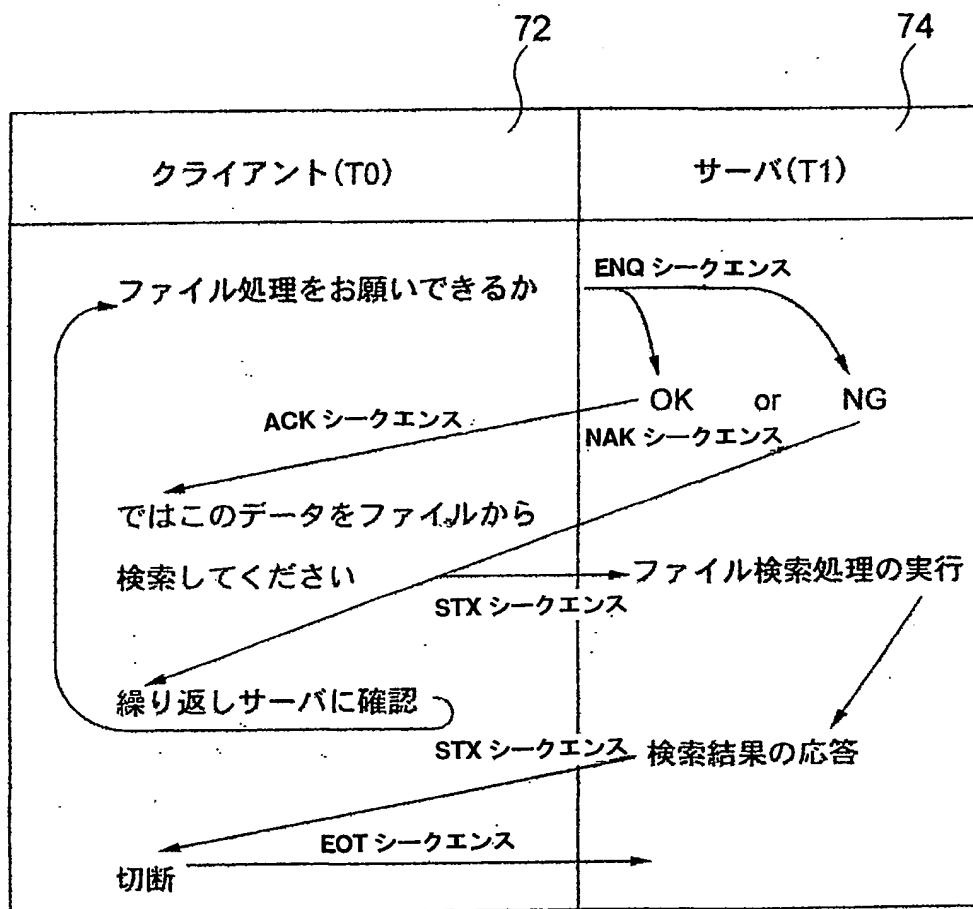
6 / 1 0

図 7



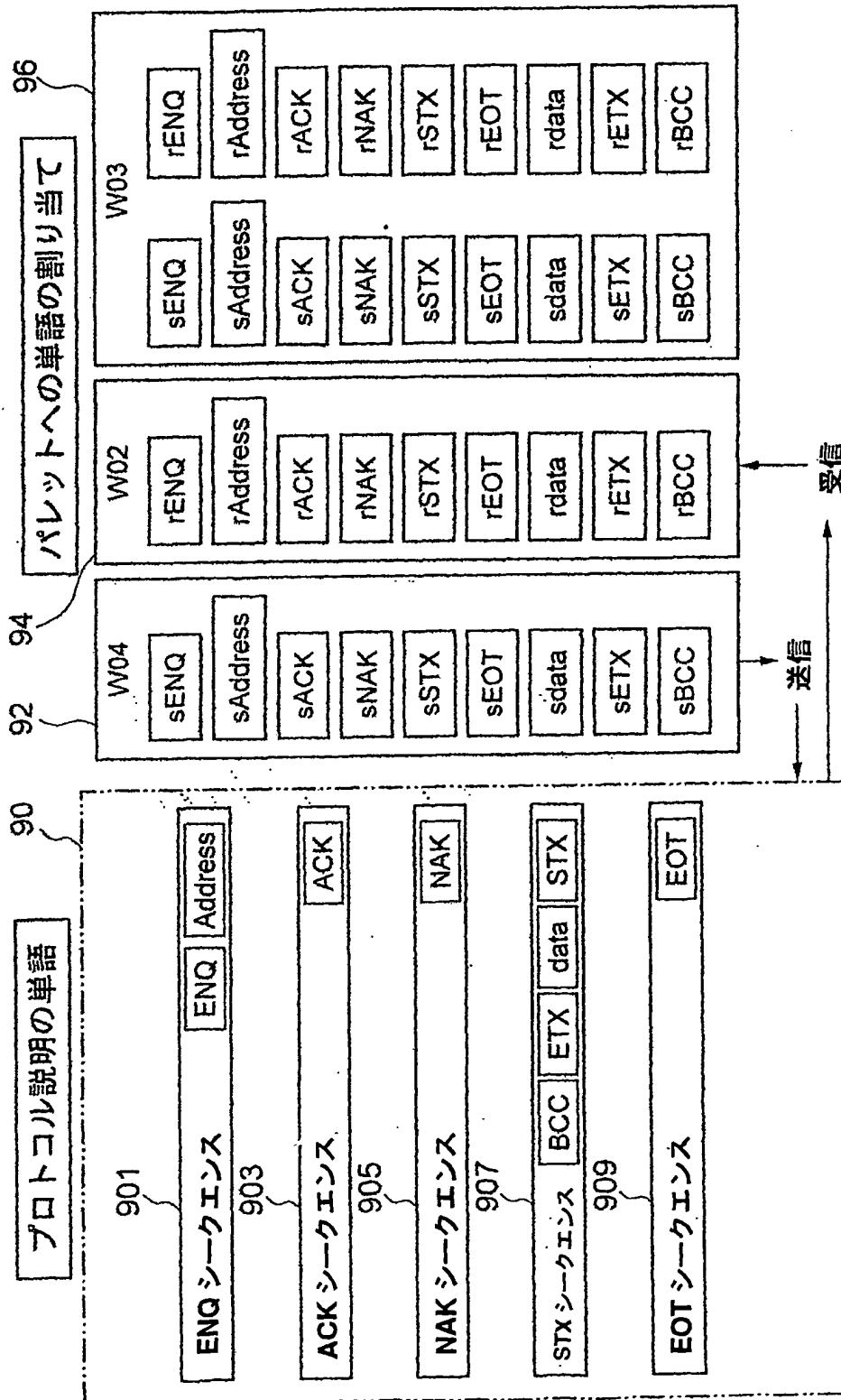
7 / 1 0

図 8



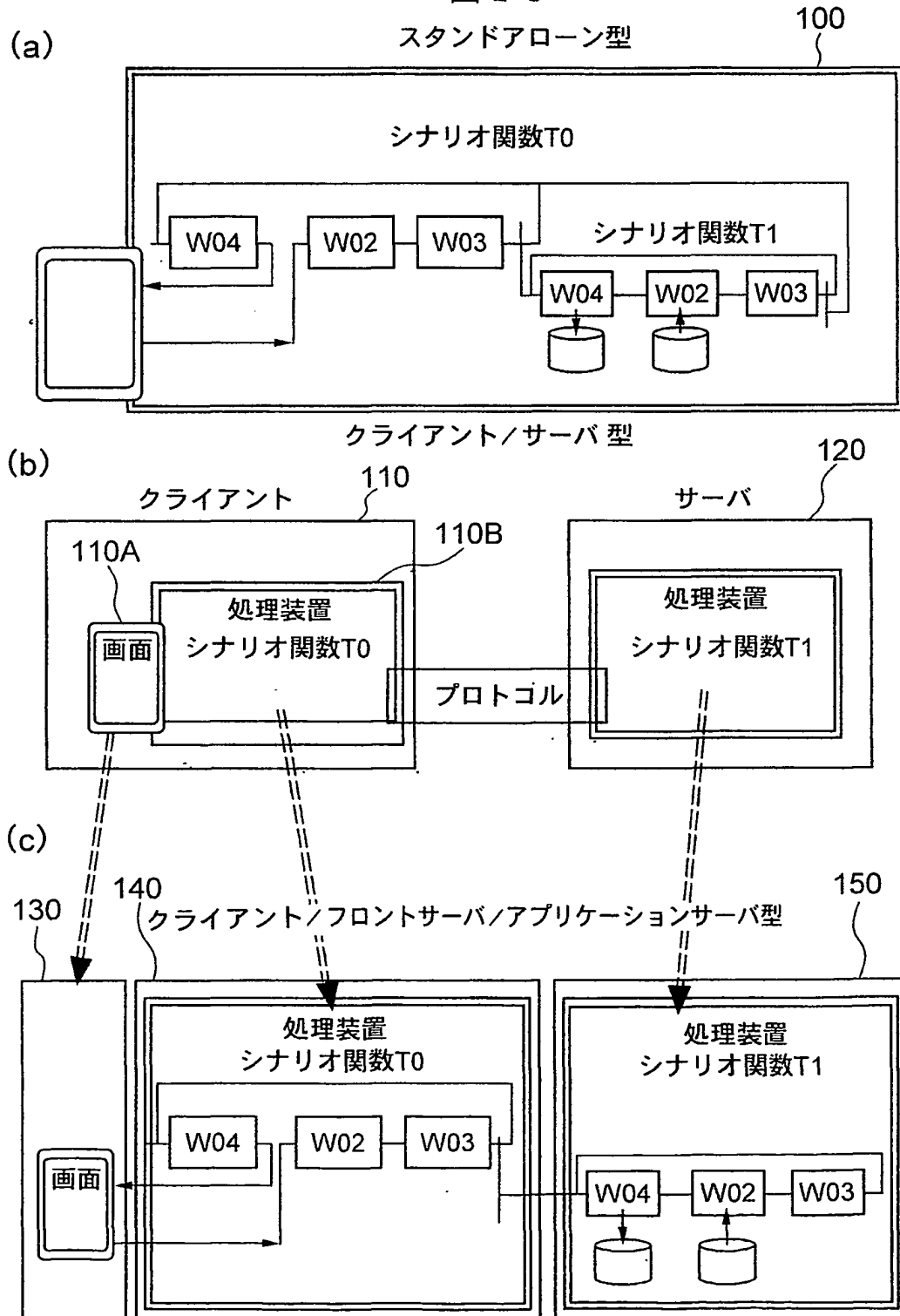
8 / 10

図 9



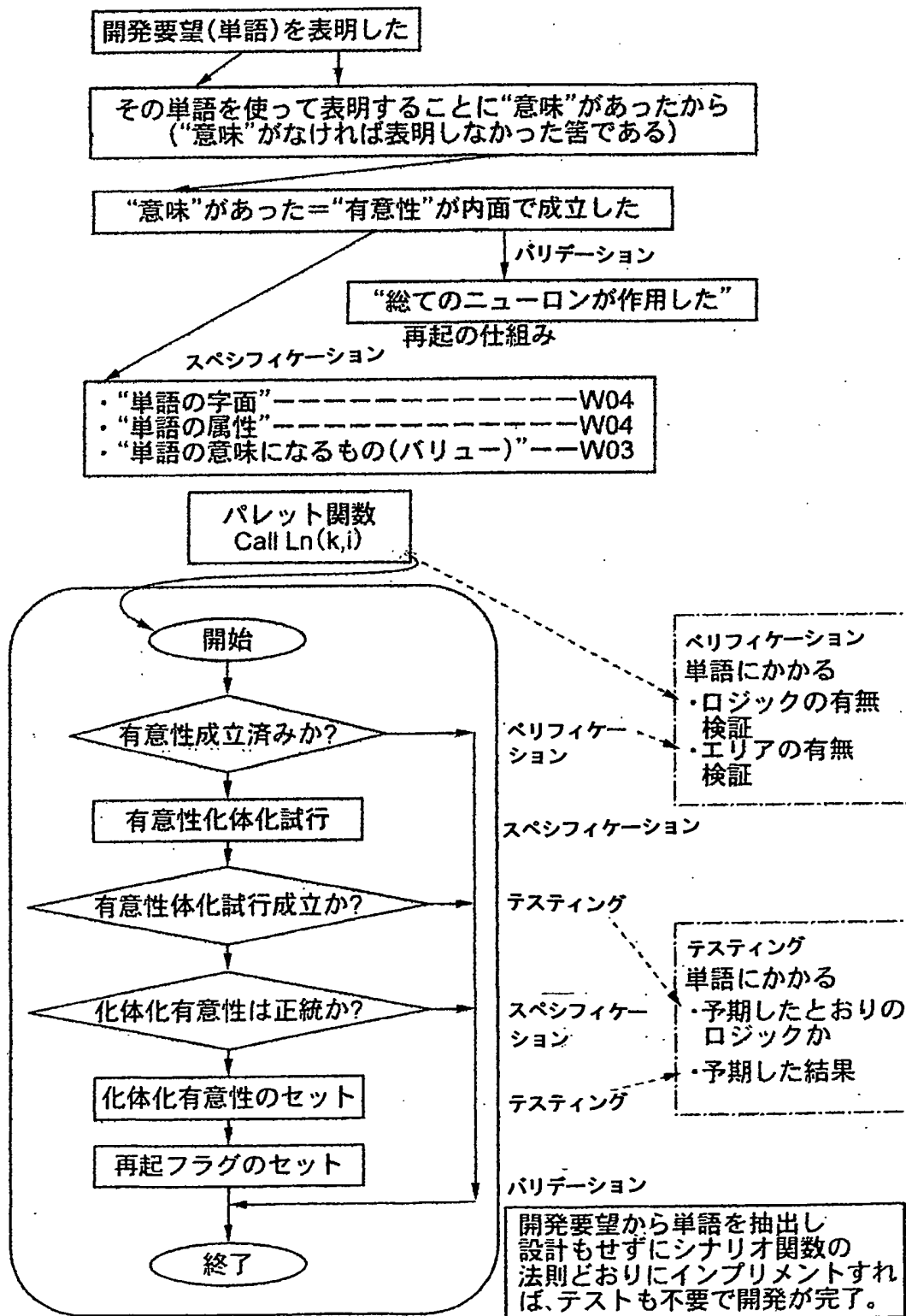
9 / 10

図 10



10/10

図 1 1



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP01/10254

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl⁷ G06F9/06, G06F9/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ G06F9/06, G06F9/44

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Jitsuyo Shinan Koho 1926-1996 Jitsuyo Shinan Toroku Koho 1996-2002
Kokai Jitsuyo Shinan Koho 1971-2002 Toroku Jitsuyo Shinan Koho 1994-2002

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 99/49387 A1 (Information System Development Institute, The Institute of Computer Based Software Methodology and Technology), 30 September, 1999 (30.09.1999), Full text; Figs. 1 to 13 & EP 1071009 A1	1-4
X	WO 98/19232 A1 (Fumio NEGRO), 07 May, 1998 (07.05.1998), Full text; Figs. 1 to 48 & US 6138268 A & EP 859313 A1 & WO 97/16784 A1 & CA 2235924 A & NZ 321670 A & EP 947916 A1	1-4

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
19 February, 2002 (19.02.02)

Date of mailing of the international search report
26 February, 2002 (26.02.02)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. C1' G06F9/06, G06F9/44

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. C1' G06F9/06, G06F9/44

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1926年-1996年

日本国公開実用新案公報 1971年-2002年

日本国実用新案登録公報 1996年-2002年

日本国登録実用新案公報 1994年-2002年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	WO 99/49387 A1 (株式会社アイエスデー研究所, ソフトウェア生産技術研究所株式会社) 1999. 09. 30, 全文, 第1-13図 & EP 1071009 A1	1-4
X	WO 98/19232 A1 (根来文生) 1998. 05. 07, 全文, 第1-48図 & US 6138268 A & EP 859313 A1 & WO 97/16784 A1 & CA 2235924 A & NZ 321670 A & EP 947916 A1	1-4

☐ C欄の続きにも文献が列举されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの

「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」 口頭による開示、使用、展示等に言及する文献

「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」 同一パテントファミリー文献

国際調査を完了した日

19. 02. 02

国際調査報告の発送日

26.02.02

国際調査機関の名称及びあて先

日本国特許庁 (ISA/J P)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

漆原 孝治

5 B

9 3 6 6

電話番号 03-3581-1101 内線 3546